Naval Surface Warfare Center
Carderock Division
Bethesda, MD 20084-5000

NSWCCARDIV-TR-95/013 March 1996

Survivability, Structures, and Materials Directorate

Technical Report

Moisture Diffusion Analysis in Multilayer Composite Materials by Finite Difference Analysis

by

J. M. Augl (NSWC, Carderock Division, Code 681) and

A. E. Berger (NSWC, Dahlgren Division, Code B44)

19960412 043

Approved for public release; distribution is unlimited.

DTIC QUALITY INSPECTED 1

NSWCCARDIV-TR-95/013 Moisture Diffusion Analysis in Multilayer Composite Materials by Finite Difference Analysis

Naval Surface Warfare Center
Carderock Division
Bethesda, MD 20084-5000

NSWCCARDIV-TR-95/013 March 1996

Survivability, Structures, and Materials Directorate

Technical Report

Moisture Diffusion Analysis in Multilayer Composite Materials by Finite Difference Analysis

by

J. M. Augl (NSWC, Carderock Division, Code 681) and

A. E. Berger (NSWC, Dahlgren Division, Code B44)

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	March 1996	Interim; 1 January 1995 - 1 January 1996

4. TITLE AND SUBTITLE
Moisture Diffusion Analysis in Multilayer Composite Materials by Finite Difference Analysis

5. FUNDING NUMBERS
SC2B;
RS34S56

6. AUTHOR(S)
Joseph M. Augl and Alan E. Berger

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Surface Warfare Center
Carderock Division
White Oak Laboratory
10901 New Hampshire Ave.
Silver Spring, MD 20903-5640

8. PERFORMING ORGANIZATION REPORT NUMBER
NSWCCARDIV-TR-95/013

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Office of Naval Research
Ballston Tower One
800 N Quincy St
Arlington, VA 22217-5660

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

This report provides a FORTRAN source code for calculating the moisture uptake, diffusion, and internal distribution through the thickness of multi-layered composite sandwich materials as a function of time. This finite difference diffusion code (FDDC) was designed as a handy tool for materials engineers and electronics engineers who have no easy access to a finite element diffusion code (FEDC), yet who have a need to study long-term moisture effects on composites or sandwich structures. A unidirectional diffusion analysis through laminated plates or sandwich panels with many different material layers, for many different environments, can be performed within a few seconds of CPU time. The code can be run on Macintosh or IBM-compatible computers. A sample run is provided which may serve as a tutorial and also as a comparison with a previously run finite element analysis of the same problems. The differences in the sample runs for the calculated moisture concentrations of the FDDC and FEDC analyses were insignificant (less than one-half percent).

14. SUBJECT TERMS
finite element analysis; finite difference analysis; composite materials; FORTRAN; moisture diffusion

15. NUMBER OF PAGES
56

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

FOREWORD

The work described herein was sponsored by the Office of Naval Research. It was administered by Mr. Ivan Caplan, Materials Block Manager at the Naval Surface Warfare Center, Carderock Division (NSWCCD), Code 0115, under the Ship Submarine Materials Program (SC2B), Composite Materials Project (RS34S56), and NSWCCD Work Unit 1-6440-613.

The purpose of this work was to provide materials engineers and electronics engineers with a versatile, easily implemented, portable computer program for predicting the uptake and internal distribution of moisture in laminated composites and sandwich materials. Different environments can easily be modeled by using an averaging procedure for obtaining kinetic average relative humidities (RH) and average temperatures. The effect of material variation and layer thicknesses can be readily studied. Neither a finite element code nor knowledge of finite element analysis is required. One may use this code for predicting moisture diffusion in these materials for a specific marine environment or for accelerated test conditions in a laboratory.

The compiled FORTRAN code and/or the source code is available via Internet e-mail or on a floppy disk by contacting one of the authors: augl@oasys.dt.navy.mil or aberger@relay.nswc.navy.mil.

This effort is a continuation of two previous reports that described the experimental determination of the constituent diffusion coefficients and solubilities of materials for specific naval applications, and the use of finite element analysis to study such moisture diffusion problems. Results of finite element analyses were compared with the finite difference model used in this report and were found to give excellent agreement.

Approved by:

Carl E. Mueller

CARL E. MUELLER, Head
Weapons Materials Department

ABSTRACT

This report provides a FORTRAN code for calculating the moisture uptake, diffusion, and internal distribution through the thickness of multi-layered composite sandwich materials as a function of time. This finite difference diffusion code (FDDC) was designed as a handy tool for materials engineers and electronics engineers who have no easy access to a finite element diffusion code (FEDC), yet who have a need to study long-term moisture effects on composites or sandwich structures. A unidirectional diffusion analysis through laminated plates or sandwich panels with many different material layers, for many different environments, can be performed within a few seconds of CPU-time. The code can be run on Macintosh or IBM-compatible computers. A sample run is provided which may serve as a tutorial and also as a comparison with a previously run finite element analysis of the same problems. The differences in the sample runs for the calculated moisture concentrations of the FDDC and the FEDC analyses were insignificant (less than one-half percent).

CONTENTS

ILLUSTRATIONS

TABLES

INTRODUCTION

Moisture can affect mechanical, and electrical properties of structures made from organic matrix composite or sandwich materials. A knowledge of the amount of moisture absorbed and its distribution as a function of time is sometimes very important. For instance, the radar transmission loss due to water absorption in sandwich materials to be used on ships may be a key factor in deciding an optimum material combination. Thus, a prediction of how much water is absorbed over the life-time of the structure is important.

Another example for the interest in such analyses is in the preparation of test items with a specified amount of sorbed moisture in as short a time as possible. To accelerate the moisture diffusion, one usually increases the temperature at constant RH value. The details for the length of exposure at elevated temperature can be calculated.

This report provides a one-dimensional moisture diffusion computer program, called DIFFUSE, for predicting the moisture uptake and internal distribution of moisture in organic matrix composite or sandwich materials where the individual layers have different diffusion coefficients. The program provides the materials engineers with a simple tool that will permit them to make quick estimates of moisture uptake and distribution as a function of time and the environment, that is, temperature and relative humidity (RH).

Such analyses are usually done with finite element analysis (FEA) codes. These codes are not always available. They are expensive, and require a substantial learning time to use them efficiently. Also, each new combination of materials and changes in their dimensions usually requires a time consuming preprocessing step for the new model. If the engineer has no direct access to such a facility and needs to rely on someone else to do the analysis, at a remote site, the turnaround time for different material combinations could be substantial.

This code that can be run on any desk-top PC or Macintosh computer with a minimum of effort. The individual material properties of the multilayered plates are entered in an input file which can be quickly modified by entering the changes in number of material layers, their dimensions and diffusion properties. There are no essential restrictions in the number of material layers or sub-layers of the sandwich one wants to analyze.

The output data are given in two formats: First, a numerical table that lists a) the moisture concentrations, as a normalized function, u, of the maximum moisture solubility for a given environment (in this report for 80% RH); b) weight percent (g/100g of dry material); c) grams per cubic centimeter, at each specified time; and d) the weight of H_2O uptake per square centimeter cross-section for each material and for the overall sandwich thickness. In the second format, results are presented in columns which permits easy import into spreadsheets or graphics software for postprocessing and graphics representations. For Windows applications, this is easily accomplished with the usual copy-paste procedures.

A source code listing of the FORTRAN diffusion code DIFFUSE, available from the authors via the internet, is given in the Appendix and can be installed in any PC with a FORTRAN compiler. We used MS-FORTRAN version 5.0. The compiled version (file diffuse.exe) together with an input file, as listed in the sample of the Appendix, can be run without the necessity of a FORTRAN compiler on any DOS based PC.

Since the MS-FORTRAN version 5.0 limits the size of the source code, we have broken it up into 3 parts: diffuse.for, initl.for, and outp.for, which can be linked for compilation, to give the execution code, diffuse.exe.

We compared the results with examples previously run with a FEA code performed on the same computer.[1] The concentration differences found between the FEA and the present finite difference analysis (FDA) were insignificant. Moreover, we were pleasantly surprised to find that the CPU times for this unidirectional finite difference diffusion analysis was about 1000 times faster than that for the same problem with the corresponding FEA model using solid elements. This is due to the large overhead for the necessary element processing.

DISCUSSION

MATERIALS

Geometry and Finite Element Model of a Sample Sandwich Material, Group A

The sketch on the right (not to scale) shows a schematic representation of a symmetric sandwich construction. It contains an outside composite face sheet, followed by a core material, a glass-epoxy copper-screen, followed by another layer of core material, another copper-screen, core, and face sheet. All layers are bonded together with NB102 adhesive, nominally 11 mils thick . We determined the sorption behavior of these materials as a function of temperature in a previous report.[2] The diffusion coefficients, solubilities and dry material densities at 22°C and 80 percent RH are listed in Table 1. For brevity, we use the following material abbreviations: gl/3113 - for E-glass/epoxy (SP Systems 3113) face sheet; gl/RTM3 - for E-glass/Dow Derakane vinyl-ester (510A), (RTM3) face sheet; gl/G10 - for E-glass/epoxy (G10) FR4 copper-screen; balsa core - for balsa wood type D57 AL6000 [nominally 7 lb./ft.^{3}, manufactured by Baltec U.S.A]; and PVC core - for PVC-foam [Klegecell II (R75), is a polyvinyl chloride (PVC) foam, of a nominal density of 4.7 lb./ft.^{3}, manufactured by Polymex Corp., Italy].

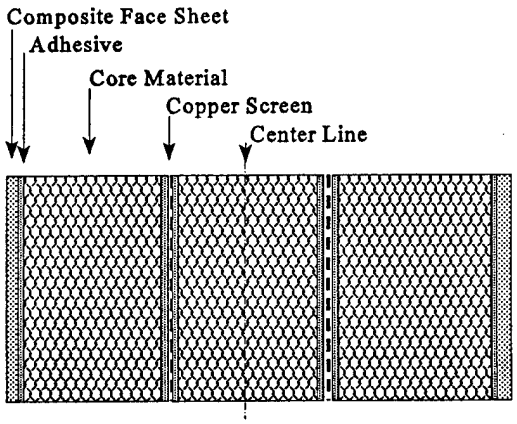

TABLE 1. Diffusion Coefficients, Solubilities and Densities of Sandwich Materials at 22°C and 80 Percent RH.

Material	Diff. Coeff.(cm^2/sec)	Solubility g/100g	Density (g/cm^3)
gl/3113	5.81E-10	0.664	1.765
gl/RTM3	2.56E-09	0.157	1.925
gl/G10	5.05E-10	0.720	1.827
balsa core	6.59E-06	14.361	0.091
PVC core	9.81E-07	1.927	0.0745

The nominal thickness dimensions for the face sheets, adhesive layers, outer core materials, FR4 copper screens, and inner core materials are: 0.1, 0.011, 1.77, 0.005 and 1.60 inches respectively. In order to use a unified set of dimensions for the diffusion coefficients, densities, and moisture solubilities, the model input and output and all constants were set up in centimeter-gram-second units. Since the adhesive layers are quite thin, the model was simplified by incorporating these layers into the face sheet and into the copper screen tape. For practical purposes, this provided reasonable estimates of what one may expect in a marine environment. A one-dimensional diffusion analysis through the thickness was performed. It was assumed that the edges of the sandwich construction were well sealed so that no moisture could diffuse from there. The result of this analysis was compared with the finite element analysis of the same problem, which was one of the sample analyses described in a previous report.[1]

ENVIRONMENT

A general approach for modeling the environment, such as encountered on ships, was described in previous reports. [1,3] Such an approach requires knowledge of the average daily temperature and humidity excursions over a long period of time. In a real environment, temperature and RH change continuously over the period of a day and with it the diffusion coefficient and the boundary conditions on the surface of the composite sandwich. However, one may use a special averaging procedure for the temperature and humidity which permits one to approach such a diffusion problem as if the humidity and temperature were kept constant. It is this kinetic average daily temperature (T_{kav}) and the kinetic average relative humidity (RH_{kav}) profile that is important for modeling the environment rather than the arithmetic average daily temperature and RH.

For a typical marine environment, we have chosen that RH_{kav} and T_{kav} correspond to 80 percent RH and 22°C respectively, which we think are reasonable for some ocean environments.[4]

MODEL DESCRIPTION

In order to obtain the moisture concentration c(x,t) (grams of H_2O/cm^3 of material) at a point x (cm) in the material at some time t (s), it is necessary to rescale the diffusion problem in view of the discontinuous behavior of the material properties across material interfaces. Let s(x) denote the equilibrium saturation of H_2O in the material at the point x (as weight percent of water = g H_2O/100 g dry material) at a given ambient relative humidity r (r is 80% within the DIFFUSE program), let D(x) be the diffusion coefficient of the material at x (cm^2/s), and let $\rho(x)$ (g/cm^3) signify the density at x (that is, the density of the dry material in the layer containing the point x). Define the scale factor k(x) to be $s(x)\rho(x)/100$, and set $u(x,t) = c(x,t)/k(x)$. For this model of moisture diffusion it is assumed that s(x) varies linearly with r, and that at an interface point z, between two material layers, the function u(x,t) is continuous across z. This is to be valid for any time t, including times before equilibrium has been reached. It is assumed that, within each material layer, D, ρ, s, and k are constant.

The flux at a given location x is then

$$F = -D(x)\frac{\partial c(x,t)}{\partial x}$$

One then has

$$F = -D(x)k(x)\frac{\partial[c(x,t)/k(x)]}{\partial x} = -D(x)k(x)\frac{\partial[u(x,t)]}{\partial x}$$

as long as x is not an interface point, since k(x) is constant within individual material layers. A mass balance analysis, detailed in the Glossary of the Appendix, shows that the diffusion equation satisfied by u(x,t) is

$$k(x)\frac{\partial u(x,t)}{\partial t} = \frac{\partial}{\partial x}[k(x)D(x)\frac{\partial u(x,t)}{\partial x}]$$

This model is appropriate as long as interface points are included in the set of finite difference grid points (finite element endpoints). It will produce numerical values for the concentration of moisture at various times within a composite material exposed at its ends (exposed surfaces) to given relative humidities.

The appropriate boundary condition at an endpoint exposed to a constant relative humidity h is u=h/80 (since s(x) is converted within DIFFUSE to the values corresponding to r=80). Concentration values c(x,t) in terms of u are c(x,t) = k(x)u(x,t). Note that at interface points, there are two values for c corresponding to the value of k(x) for the material on the left and right sides of the interface. This is the reason the problem was reformulated in terms of the single valued function u.

The program DIFFUSE solves this diffusion problem using a finite difference method corresponding to the finite element method with linear elements [Strang][5], [Wood][6], and employs tridiagonal matrix factorization [Isaacson & Keller][7] for the resulting linear systems of equations.

EXAMPLES

The moisture diffusion and internal distribution were calculated using both, FEA and FDA for two examples of sandwich structures, each with the same geometry but different materials, as described in the Materials section. Since the diffusion equation is solved using finite elements or finite differences, one needs to investigate the effect of the number of elements or finite difference grid points, as well as the number of time steps and type of time differences used in the solution scheme.

The symmetric sandwich structures had seven layers, consisting of composite face sheet/core/copper-screen/core/copper-screen/core/face sheet with nominal thicknesses of 0.1/1.77/0.0158/1.60/0.0158/1.77/0.1 inches respectively. These layers are bonded together with an adhesive nominally 0.011 inch thick. For the purpose of modelling the adhesive is considered part of the face sheet and screen.

The material properties (diffusion coefficients and solubilities at 22°C and 80 percent RH, and their dry densities) are listed in Table 1.

The source code of the finite difference scheme used here is listed in the Appendix, starting on page A-4.

User instructions for the inputs for each distinct layer is given on page A-23 of the Appendix (see also A-1/A-2 and the Glossary).

The input listing, inp5, for a sample run is given on page A-24 of the Appendix.

The corresponding output listings, out5 and out5col, start on page A-25. In order to import these data for graphing with the use of a spread sheet or a graphics application, such as SigmaPlot®, out5col gives the same information in a columned output (starting on page A-32). Note: the number of output points per material layer is chosen automatically. It will be five if the number of finite difference subintervals chosen for the layer is evenly divisible by four.

The table on page A-3 of the Appendix compares the results of the FEA and FDA calculations for various time steps and finite difference intervals. F, C, and S in column 3 indicate face sheet, core, and copper-screen respectively, and the number which follows the letter indicates the number of sub-layers (this is the number of finite difference subintervals or finite elements within each material layer). For simplicity, we compared the relative moisture concentrations obtained from the FEA and FDA calculations at only two points: the center point of the outer core layer, and the center point of the inner core, which is also the center of the sandwich. The x-coordinates of these points are respectively 2.53 and 6.906 cm into the sandwich.

Column four lists the time in months for which the relative moisture concentrations are given at these two points (columns five and six). The relative moisture concentrations listed in the table are the percentage of the maximum moisture concentration that the specified material can absorb at a given temperature and at a specified RH (here 22°C and 80 percent RH).

Note: By using relative moisture concentrations, the values just to the left and right of the interface are the same. However, if one plots the concentrations in weight percent instead of relative percent, one observes that there is actually a discontinuity of moisture concentration at the interface from one material to another. The reason for this is that, at equilibrium, different materials usually have different solubilities. Therefore, the concentration of moisture at the left side of the interface is different from that of the right side. To convert the values of relative moisture concentration to weight percent moisture, one multiplies the relative concentrations by the maximum moisture uptake and divides by 100. Columns seven and eight list the concentration in weight percent for the respective positions of 2.53 and 6.906 cm into the sandwich. It is

therefore not surprising that the core (in this case balsa wood or PVC-foam) can contain considerably higher concentration of moisture than the face sheet.

As one can see from the listed relative moisture concentrations, the difference between using 1- or 4-months time steps for the analysis is only on the order of 10^{-3} percent, and the difference between the finite element and finite difference method is, for the worst case (240 months), only 0.56 percent. Even when the spatial differences were made much smaller, as in the example with (F16/C80/S12/C80/S12/C80/F16), the concentration differences are seen only in the fourth decimal place, nor was there a significant difference when the spatial differences were made larger, as in the example with (F4/C4/S4/C8/S4/C4/F4).

To determine concentrations after only a few time steps, one should check for oscillations in the solution which are artifacts. We recommend that first one carry out a few runs where the time and/or space intervals are reduced. Such oscillatory artifacts, observed in the short-time solution, can be eliminated if "sufficiently small" time steps are chosen (refer to the discussion on the bottom of page A-19). These initial oscillations usually die out and have no effect in the accuracy of the long term solutions.

Figures 1 and 2 compare the results of the FEA [1] and of the FDA for a sandwich with gl/3113 epoxy face sheet, balsa core, and gl/G10 epoxy embedment layer (but without the copper screen). Figures 3 and 4 show the equivalent results with gl/RTM3 vinyl face sheet, PVC core, and gl/G10 epoxy embedment layer (with 50 percent copper cross-section). They are, for all practical purposes, superposable. The FEA and FDA results differ by less than 0.6 percent.

A glossary of the variable names in the program DIFFUSE together with a detailed explanation of the algorithm used for the program begins on page A-16 of the Appendix. For practical use and distribution of the code, we have commented each line of the glossary with "c" so that the glossary could be part of the FORTRAN code.

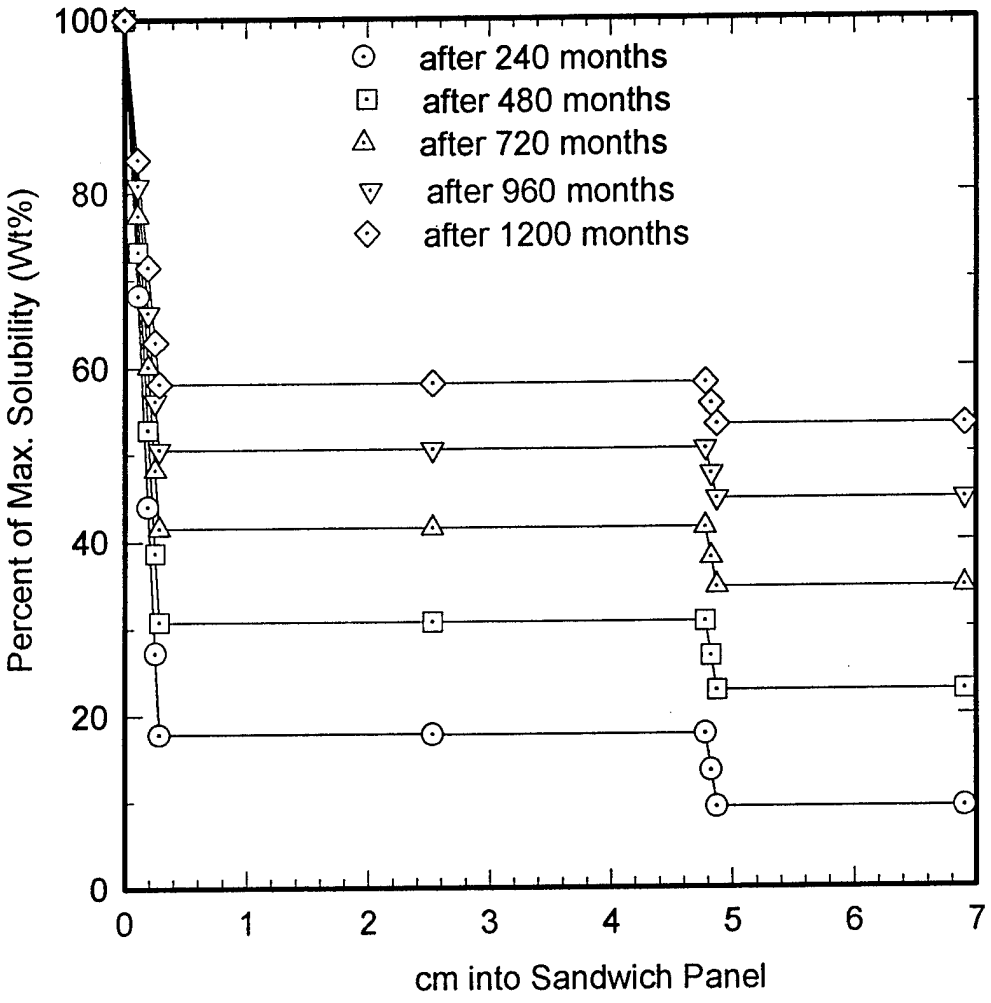

Figure 1. Relative Moisture Concentration in Glass/3113 Epoxy Balsa Wood Sandwich Panel (By FEA).

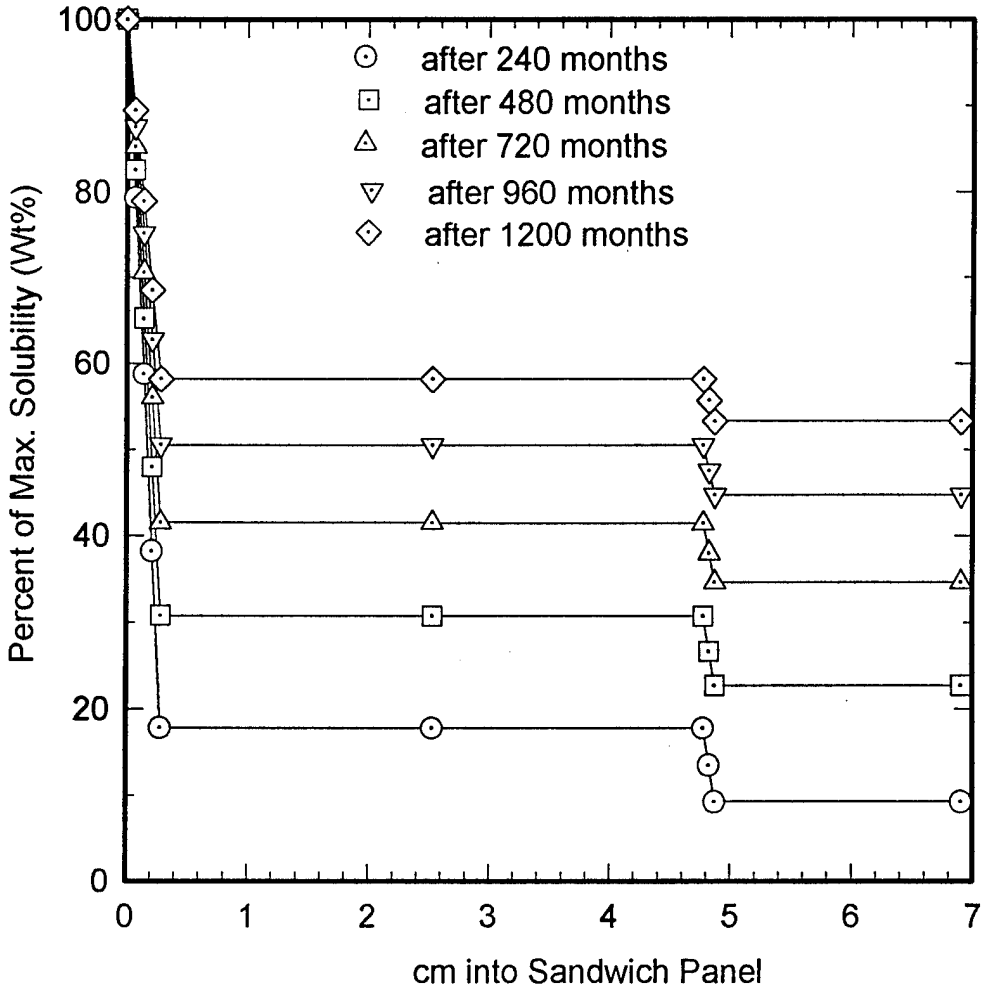

Figure 2. Relative Moisture Concentration In Glass/3113 Epoxy Balsa Wood Sandwich Panel (By FDA).

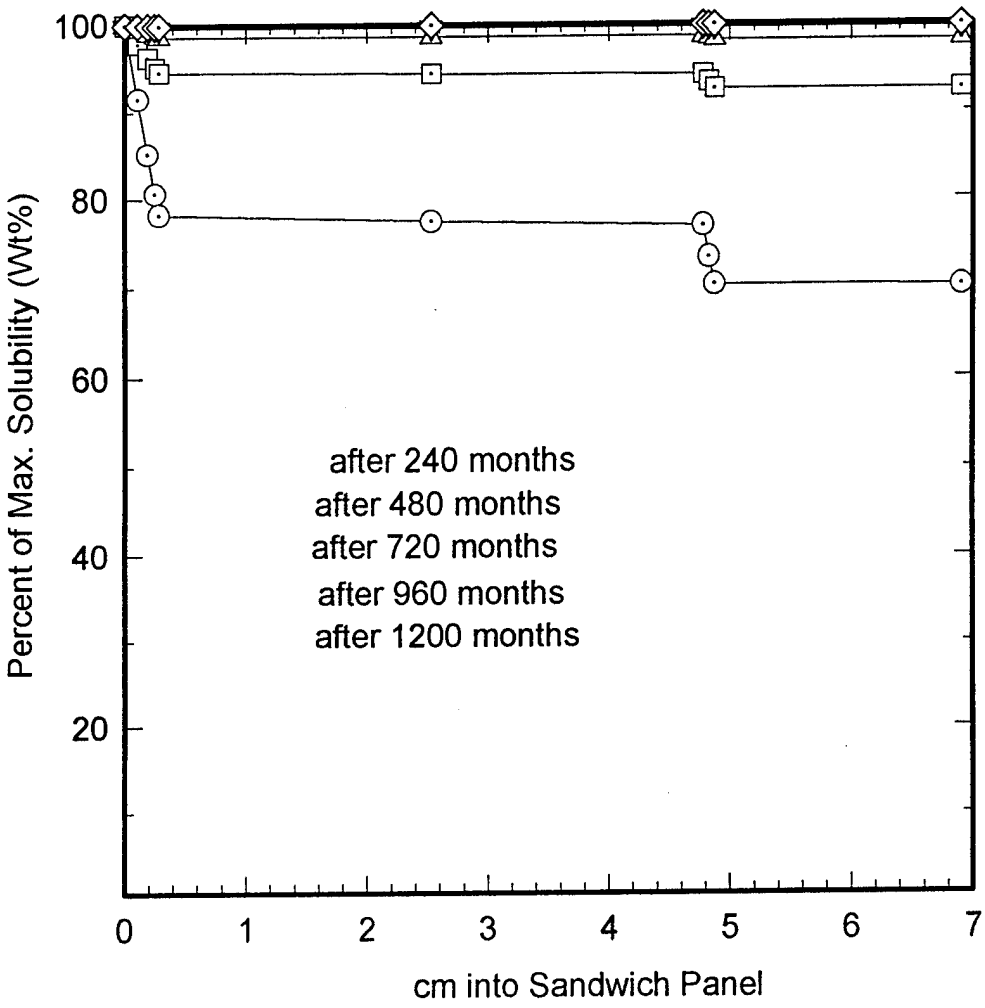

Figure 3. Relative Moisture Concentration in Glass/Rtm3 Vinyl Ester Pvc-Foam Sandwich Panel Including a 50 Percent Copper Screen Cross Section (By FEA).

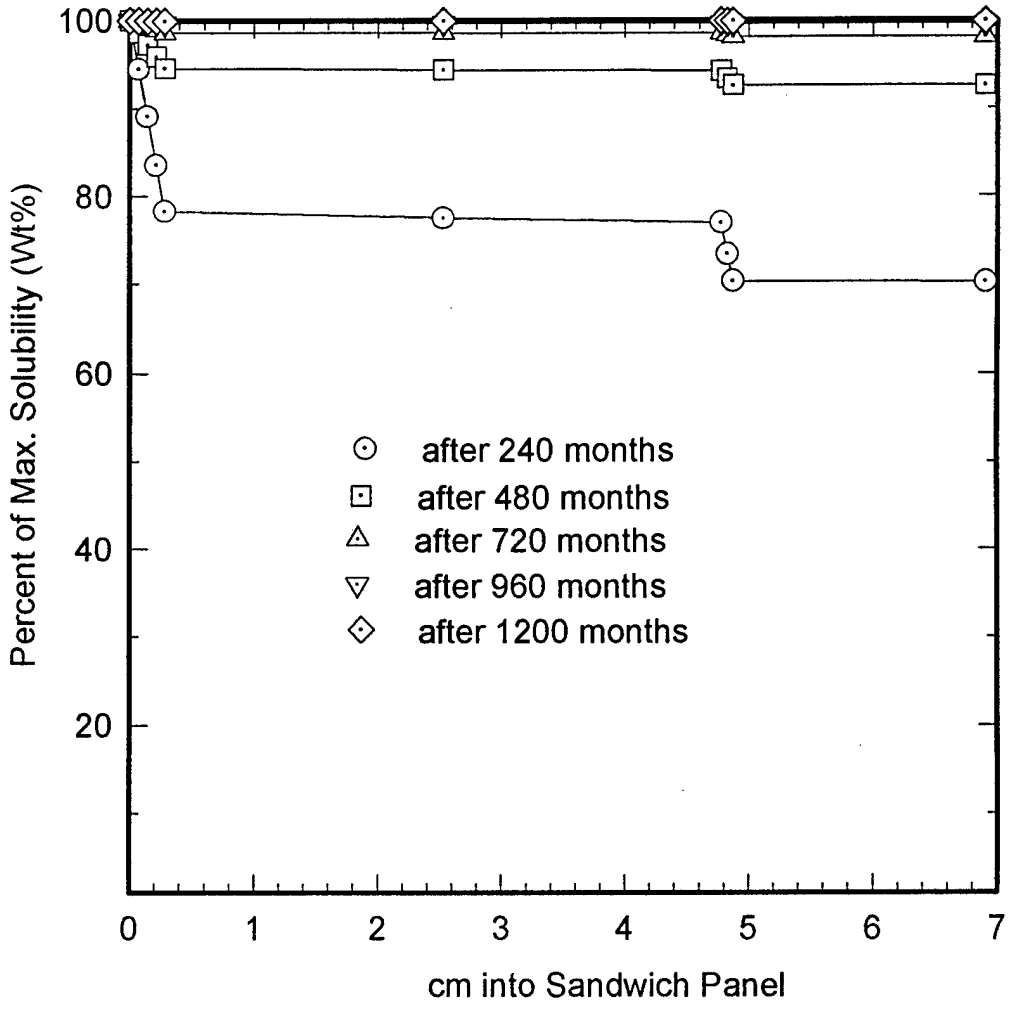

Figure 4. Relative Moisture Concentration in Glass/Rtm3 Vinyl Ester Pvc-Foam Sandwich Panel Including a 50 Percent Copper Screen Cross Section (By FDA).

CONCLUSIONS

A one-dimensional finite difference analysis code for calculating moisture uptake, diffusion, and internal distribution in composite laminates or sandwich materials is provided in the Appendix. The main purpose was to give the materials engineer or electronics engineer a simple tool that can be readily used on a desk-top computer to calculate diffusion problems without having to invest in learning and acquiring an expensive finite element code.

If the user has no FORTRAN compiler available, a compiled version of the code can be used. This requires only the input file for the material dimensions (number of layers, thicknesses and number of sub-layers), material properties (densities, diffusion coefficients, and maximum moisture solubilities at specified RH values) and boundary conditions (left and right outside RH values) for specific laminates or sandwiches. Since this code is one-dimensional, it will only solve problems where the edge effects can be neglected, that is, where the edges are either sealed or where the aspect ratio of length to plate thickness is sufficiently large.

For tutorial purposes, we have provided specific examples for a sandwich previously analyzed using a FEA.[1] The time steps for these examples were one and four months. The differences in the FEA and FDA results after 240, 480, 720, 960, and 1200 months were less than 0.6 percent. The differences in CPU times for the same problems run with the finite difference code and the finite element code was 1:1000. For one-dimensional problems the finite difference code is therefore preferable.

These calculations assume the densities, maximum moisture solubilities, and diffusion coefficients of all constituent materials are known as a function of temperature and of RH. In this way, a model of various environmental conditions can be approximated with kinetic average diffusion coefficients (D_{kav}) and kinetic average RH values (RH_{kav}).

The kinetic average diffusion coefficient was chosen to be that of 22°C and the kinetic average RH_{kav} to be 80 percent which are reasonable for some ocean environments.[4]

REFERENCES

1. Augl, J. M., *Use of Finite Element Analysis for Transient Moisture Diffusion Studies in Multilayer Composite AEM/S System Sandwich Materials,* CARDIVNSWC/TR-94/019, November 1994.

2. Augl, J. M., *Sorption and Diffusion of Moisture in Multilayer Composite AEM/S System Sandwich Material,* CDNSWC/TR-94/018, October 1994.

3. Augl, J. M. and Berger, A. E., *The Effect of Moisture on Carbon Fiber Reinforced Epoxy Composites. III Prediction of Moisture Sorption in a real Outdoor Environment,* NSWC/WOL/TR 77-13, June 1977.

4. *Engineering Design Handbook: Environmental Series Part Two: Natural Environmental Factors,* Army Materials Command, AMCP # 706-116, p(11-17), April 1975.

5. Strang, G. S., *Introduction to Applied Mathematics,* Wellesley-Cambridge Press, Wellesley MA p(571-576), 1986.

6. Wood, W. L., *Practical Time-stepping Schemes,* Oxford U. Press, 1990.

7. Isaacson, E. and Keller, H. B., *Analysis of Numerical Methods,* John Wiley & Sons, Inc., NY, p(55-58), 1966 .

APPENDIX

Page A-3 lists abbreviated results of an example of finite element and finite difference analyses where the time steps for the analyses and the number of finite elements and finite differences were varied for the same sandwich geometry of a balsa wood and a PVC core material. The different material layers in the sandwich material are separated by the "/" symbol, and the number of sublayers for the FE or FD models are indicated with numerals. For instance, the sandwich indicated by (F4/C20/S4/C20/S4/C20/F4) consists of a sequence of four face sheet sublayers (=finite elements or finite differences), 20 core sublayers, 4 copper-screen-sublayers, 20 core sublayers, 4 copper-screen-sublayers, 20 core sublayers, and 4 face sheet sublayers. While the material layers are adhesively bonded together, the sublayers have no material boundaries; they only serve for the mathematical solution scheme. The finite element analysis carried out as described in the previous report [1] took advantage of the nonuniform size of subelements and of symmetry conditions, requiring only half as many elements. For an easy comparison of the FEA and FDA results, the relative and weight percent moisture concentrations are only listed at two points: x = 2.53 and at 6.906 cm into the panel. These points are at the center of the first core layer, and at the center of the inner core layer. Time intervals were 240, 480, 720, 960, and 1200 months, where the chosen time steps, dt, were 1 and 4 months). Also included in this table are FDA results with 4 times the number of sublayers, and with a dt of 0.5 month, and fewer sublayers.

The remainder of this appendix is organized as follows:

The present setting of the parameters in COMINCL which give the array dimensions in DIFFUSE are such that there can be up to 20 different material layers (nlmax=20). The maximum allowed total number of finite difference subintervals used in DIFFUSE to subdivide all the material layers is one less than the maximum number of finite difference points permitted, which is 800 (in COMINCL nptsmax=800). The parameters nlmax and nptsmax, defined in the first line of

COMINCL, can be increased if more layers are necessary.

We have included an example for a sandwich construction previously calculated using a FEA code. The program instructions and input instructions are given on page A-23. The input file was called inp5. The output is presented in two different ways which are shown on pp. A-25 and A-32. The column output (p. A-32) is useful for graphical representation. These columns can be readily imported into spreadsheets or graphics applications such as SigmaPlot®.

Relative Moisture Concentration at 22 Deg.C and 80 percent RH in various Sandwich Materials							
Balsa Wood Core, 3113 Glass-Epoxy Face Sheet Material, FR4 Embedment Layer without Cu							
Analysis	**Time Steps**	**Material#Sublayers/**	**Time (Mo)**	**at x1=2.53**	**at x2=6.906**	**Wt% at x1**	**Wt% at x2**
FEA	1 Month	F4/C20/S4/C20/S4/C20/F4	240	17.831	9.3461	2.561	1.342
			480	30.783	22.808	4.421	3.275
			720	41.491	34.676	5.959	4.980
			960	50.525	44.756	7.256	6.427
			1200	58.162	53.286	8.353	7.652
	4 Months	F4/C20/S4/C20/S4/C20/F4	240	17.831	9.3459	2.561	1.342
			480	30.761	22.807	4.418	3.275
			720	41.491	34.676	5.959	4.980
			960	50.525	44.756	7.256	6.427
			1200	58.162	53.284	8.353	7.652
FDA	1 Month	F4/C20/S4/C20/S4/C20/F4	240	17.789	9.2939	2.555	1.335
			480	30.749	22.753	4.416	3.268
			720	41.462	34.629	5.954	4.973
			960	50.499	44.715	7.252	6.422
			1200	58.14	53.248	8.349	7.647
	4 Months	F4/C20/S4/C20/S4/C20/F4	240	17.789	9.2937	2.555	1.335
			480	30.749	22.753	4.416	3.268
			720	41.462	34.629	5.954	4.973
			960	50.499	44.715	7.252	6.422
			1200	58.14	53.248	8.349	7.647
	1 Month	F16/C80/S12/C80/S12/C80/F16	240	17.758	9.2654	2.550	1.331
			480	30.725	22.726	4.412	3.264
			720	41.441	34.606	5.951	4.970
			960	50.482	44.695	7.250	6.419
			1200	58.117	53.231	8.346	7.645
	0.5 Months	F4/C4/S4/C8/S4/C4/F4	240	17.789	9.2936	2.555	1.335
			480	30.749	22.753	4.416	3.268
			720	41.462	3.4629	5.954	0.497
			960	50.499	44.715	7.252	6.422
			1200	58.14	53.247	8.349	7.647
PVC Core, RTM3 Glass-Vinyl Face Sheet Material, FR4 Embedment Layer with 50 percent Cu Cross-Section							
Analysis	**Time Steps**	**Material#Sublayers/**	**Time (Mo)**	**at x1=2.53**	**at x2=6.906**	**Wt% at x1**	**Wt% at x2**
FEA	1 Month	F4/C20/S4/C20/S4/C20/F4	240	77.432	70.149	11.120	10.074
			480	94.333	92.504	13.547	13.284
			720	98.577	98.118	14.157	14.091
			960	99.643	99.527	14.310	14.293
			1200	99.91	99.881	14.348	14.344
	4 Months	F4/C20/S4/C20/S4/C20/F4	240	77.434	70.15	11.120	10.074
			480	94.334	92.504	13.547	13.284
			720	98.577	98.118	14.157	14.091
			960	99.643	99.527	14.310	14.293
			1200	99.91	99.881	14.348	14.344
FDA	1 Month	F4/C20/S4/C20/S4/C20/F4	240	77.476	70.175	11.126	10.078
			480	94.357	92.528	13.551	13.288
			720	98.586	98.128	14.158	14.092
			960	99.646	99.531	14.310	14.294
			1200	99.911	99.883	14.348	14.344
	4 Months	F4/C20/S4/C20/S4/C20/F4	240	77.478	70.177	11.127	10.078
			480	94.358	92.529	13.551	13.288
			720	98.587	98.128	14.158	14.092
			960	99.646	99.533	14.310	14.294
			1200	99.911	99.883	14.348	14.344

FORTRAN SOURCE FILE: cominc1

```
      parameter(nlmax=20, nptsmax=800, kmax=nptsmax+nlmax)
      implicit double precision (a-h,o-z)
      common /comr/ rhsol, xrt, thicka(nlmax),
     &  diffcfa(nlmax), rhoa(nlmax), outrhl, outrhr, celeft, cert,
     &  scalea(nlmax), solwpa(nlmax), xpta(0:nptsmax),
     &  dxa(nptsmax), ua(0:nptsmax), ca(kmax), xa(kmax), wtpera(kmax),
     &  h2olra(nlmax), totalw, t, tfinal,  tf, tout, toutfac, dt, alpha,
     &  al(nptsmax), ad(nptsmax), ar(nptsmax), rtside(nptsmax),
     &  aln(nptsmax), adn(nptsmax), arn(nptsmax), scra(nptsmax),
     &  zero, one, two
      character chartin*1, infile*20, outfile*20
      common /comc/ chartin, infile, outfile
c integers are integer*4
      common /comi/ nlayers, nintlra(nlmax), nint, npts, nptsm1,
     &  nptsm2, nunkn, ktop, ntsteps, kdt, levt, initu, krtbc, kleftbc,
     &  isymrt, lumping
```

FORTRAN SOURCE FILE: diffuse.for

```
      program diffuse
      include 'comincl'
      npppsi = 4
c in outp, for each material layer print out values at npppsi+1 points
      call initl
      call setup
      call outp(npppsi)
      do 100 index = 1, ntsteps
         kdt = index
         t = t + dt
         call advance
         iscr = kdt / levt
         iscr = iscr * levt
         tout = t/toutfac
c        if(kdt .eq. (levt/20)) call outp(npppsi)
c above call to output to monitor ''short time'' behavior of numerical
c solution
         if(iscr .eq. kdt) call outp(npppsi)
100   continue
      close(66)
      close(67)
      stop'normal stop in diffuse'
      end
      subroutine advance
c advance the solution one dt time step
c have matrix  al, ad, ar  of tridiagonal linear system
c form right hand side of current (nth time step) solution
c present value of t is at ''new'' time level (n+1 time level) where
c are about to calculate solution values
      include 'comincl'
      onema = one - alpha
      three = 3.d0
      six = 6.d0
      k = 1
      x = xpta(k)
      told = t - dt
      ubdyn = dirl(told)
      ubdynpl = dirl(t)
      if(initu .eq. 1) ubdyn = truesol(zero,told)
      if(initu .eq. 1) ubdynpl = truesol(zero,t)
      rtside(k) = aln(k)*ubdyn + adn(k)*ua(k) + arn(k)*ua(k+1)
      rtside(k) = rtside(k) - al(k)*ubdynpl
      if(initu .eq. 1) then
         xl = xpta(k-1)
         xr = xpta(k+1)
         hk = dxa(k)
         hkp = dxa(k+1)
         hk6 = hk/six
         hk3 = hk/three
         hkp6 = hkp/six
         hkp3 = hkp/three
c        level n+1 terms
         sl = source(xl,t)
         sd = source(x,t)
         sr = source(xr,t)
         rtside(k) = rtside(k)+alpha*(sl*hk6 + sd*(hk3+hkp3) + sr*hkp6)
c        level n terms
         sl = source(xl,told)
         sd = source(x,told)
         sr = source(xr,told)
         rtside(k) = rtside(k)+onema*(sl*hk6 + sd*(hk3+hkp3) + sr*hkp6)
      end if
c
      nunknm1 = nunkn - 1
      do 100 k = 2, nunknm1
         x = xpta(k)
         rtside(k) = aln(k)*ua(k-1) + adn(k)*ua(k) + arn(k)*ua(k+1)
```

```
            if(initu .eq. 1) then
            xl = xpta(k-1)
            xr = xpta(k+1)
            hk = dxa(k)
            hkp = dxa(k+1)
            hk6 = hk/six
            hk3 = hk/three
            hkp6 = hkp/six
            hkp3 = hkp/three
c           level n+1 terms
            sl = source(xl,t)
            sd = source(x,t)
            sr = source(xr,t)
            rtside(k) = rtside(k)+alpha*(sl*hk6 + sd*(hk3+hkp3) + sr*hkp6)
c           level n terms
            sl = source(xl,told)
            sd = source(x,told)
            sr = source(xr,told)
            rtside(k) = rtside(k)+onema*(sl*hk6 + sd*(hk3+hkp3) + sr*hkp6)
            end if
100     continue
c
        k = nunkn
        x = xpta(k)
        ubdyn = dirr(told)
        ubdynpr = dirr(t)
        if(initu .eq. 1) ubdyn = truesol(xrt,told)
        if(initu .eq. 1) ubdynpr = truesol(xrt,t)
        rtside(k) = aln(k)*ua(k-1) + adn(k)*ua(k) + arn(k)*ubdyn
        rtside(k) = rtside(k) - ar(k)*ubdynpr
        if(initu .eq. 1) then
            xl = xpta(k-1)
            xr = xpta(k+1)
            hk = dxa(k)
            hkp = dxa(k+1)
            hk6 = hk/six
            hk3 = hk/three
            hkp6 = hkp/six
            hkp3 = hkp/three
c           level n+1 terms
            sl = source(xl,t)
            sd = source(x,t)
            sr = source(xr,t)
            rtside(k) = rtside(k)+alpha*(sl*hk6 + sd*(hk3+hkp3) + sr*hkp6)
c           level n terms
            sl = source(xl,told)
            sd = source(x,told)
            sr = source(xr,told)
            rtside(k) = rtside(k)+onema*(sl*hk6 + sd*(hk3+hkp3) + sr*hkp6)
        end if
c
        call trisol(al,ad,ar,nunkn,rtside,ua(1))
        ua(0) = ubdynpl
        if(isymrt .eq. 0) ua(nunkn+1) = ubdynpr
        return
        end
        function dirl(time)
        include 'comincl'
        dirl = outrhl / 80.d0
        return
c
        entry dirr(time)
        dirr = outrhr / 80.d0
        return
        end
        function truesol(xx,time)
        implicit double precision(a-h,o-z)
c       truesol = 8.d0 * time * time + 7.d0 * xx * time + 5.d0 * xx * xx
        pid2 = 1.57079632679d0
        if(xx .lt. pid2) then
            truesol = sin(xx) * exp(-50.d0*time)
        else
            truesol = sin(5.0d0*xx) * exp(-50.d0*time)
```

```
      end if
      return
c
      entry source(xx,time)
c     source = 16.d0 * time + 7.d0 * xx - 30.d0
c     source = source * .81d0
      source = 0.d0
      return
      end
```

FORTRAN SOURCE FILE: initl.for

```
      subroutine initl
c read in data from input file (physical constants and specifications
c for the finite difference scheme)
      include 'comincl'
      character comment*72
      character cscr*3
      character outcol*23
      zero = 0.d0
      one = 1.d0
      two = 2.d0
      write(*,*)'type in file name of the input data'
      write(*,*)'(up to 20 characters  do NOT enclose in apostrophes)'
      write(*,*)'***LIMIT length of name to machine limit'
      write(*,*)'***do NOT use any leading blanks'
      read(*,500) infile
500   format(a20)
      write(*,*)'type in file name for the output'
      write(*,*)'(up to 20 characters  do NOT enclose in apostrophes)'
      write(*,*)'***LIMIT length of name to machine limit - 3'
      write(*,*)'***do NOT use any leading blanks'
      write(*,*)'note column format output file is also created'
      write(*,*)'its file name = <output file name>col'
      read(*,500) outfile
      open(55,file=infile)
      open(66,file=outfile)
      write(66,*)' output file name is ',outfile
      write(66,*)' input file name is ',infile
c get outcol name
      if(outfile(1:1) .eq. ' ') then
         write(*,*)'WARNING first char of outfile is blank, outcol=col'
         outcol = 'col'
         goto 11
      end if
      do 1000 index = 2,20
         if(outfile(index:index) .eq. ' ') then
            outcol = outfile(1:index-1)//'col'
            goto 11
         end if
1000  continue
      outcol = outfile//'col'
11    continue
      open(67,file=outcol)
      write(67,*)' column output file name is ',outcol
      write(67,*)' input file name is ',infile
      read(55,*) nlayers
      write(66,*) ' number of material layers is ', nlayers
c check dimensions
      if(nlayers .gt. nlmax) stop'nlayers gt nlmax  increase nlmax'
      read(55,*) rhsol
      write(66,*) ' rhsol = relative humidity in % at which the'
      write(66,*) ' equilibrium solubility (wt %) of H2O in all the'
      write(66,661)' material layers was measured; rhsol (%) = ', rhsol
661   format(a,f6.2)
      rhfac = 80.d0/rhsol
c use rhfac to convert everything to case where rhsol = 80% as is
c case for original code
      do 10 k=1,nlayers
         write(66,*) '          layer number ',k
         read(55,*) thick
         read(55,*) diffcf
         read(55,*) rho
         read(55,*) solwp
         thicka(k) = thick
         diffcfa(k) = diffcf
         rhoa(k) = rho
         solwpa(k) = solwp
         write(66,*) thicka(k), '  thicka (cm)'
         write(66,*) diffcfa(k), '  diffcfa (cm*cm/sec)'
```

```
          write(66,*) rhoa(k), '  rhoa (g/cc)'
          write(66,*)solwpa(k),'  solwpa(wt% @rhsol%RH gH2O/100g dry m)'
c convert to case where rhsol is 80%
          solwpa(k) = solwpa(k) * rhfac
10      continue
        isymrt = 0
        write(66,*) isymrt,  '  if 0 (1) right end is exposed (sealed)'
        kleftbc = 0
c Dirichlet boundary data at left endpoint (solution value specified)
        krtbc = isymrt
c krtbc = 0 for Dirichlet data at right endpoint,
c krtbc = 1 for symmetry condition (0 Neumann data) at right endpoint
        read(55,*) outrhl, outrhr
        write(66,*) outrhl, outrhr,  '  outside relative humidity (%)'
        write(66,*) '  at the left, right boundary (end of the material)'
        npscr = 0
        do 20 k=1,nlayers
        read(55,*) nintlr
        nintlra(k) = nintlr
        write(66,*) nintlra(k),'  # finite diff subintervals in layer ',k
        npscr = npscr + nintlr
20      continue
        npscr = npscr + 1
c check dimensions
        if(npscr .gt. nptsmax) stop'npts gt nptsmax   increase nptsmax'
        read(55,*) chartin
        cscr = '  '//chartin
        write(66,*) cscr,'  char var of length 1 --- input time units'
        write(66,*)'  time units are: s = seconds, i = minutes, '
        write(66,*)'  time units are: h = hours, d = days, w = weeks, '
        write(66,*)'  time units are: m = months (30.43667 days/month)'
        write(66,*)'  time units are: y = years (365.24 days/year)'
        read(55,*) tf
        write(66,*)tf,'  final time (in given units) calc carried out to'
        read(55,*) ntsteps
        write(66,*) ntsteps, '  # of finite diff time steps to get to tf'
        read(55,*) levt
        write(66,*)levt,' print approx soln values every levt time steps'
        read(55,*) initu
        write(66,*) initu,'  if 0 (1) start with 0 (known test) concen.'
        read(55,*) alpha
        write(66,*)alpha,'  time ave parameter, 1. = implicit, .5 = C.N.'
        read(55,*) lumping
        write(66,*) lumping, '  if 1 lump mass matrix, else do not lump'
c any remaining lines (up to 5) in input file are comments to be
c printed in the output file
        do 30 k=1,5
        read(55,501,end=31) comment
501     format(a72)
        write(66,601) comment
601     format(' ',a72)
30      continue
31      continue
        close(55)
        return
        end
        subroutine tridec(al,ad,ar,nrows)
        implicit double precision (a-h,o-z)
        dimension al(nrows), ad(nrows), ar(nrows)
c decomposes a tridiagonal matrix a into l*u
c l lower triangular (tridiagonal) with all ones on diagonal
c u upper triangular (tridiagonal)
c when  calling tridec should have  al(i)=a(i,i-1)  ad(i)=a(i,i)
c    ar(i)=a(i,i+1)
c u  is put into ad, ar
c l is put into al  (one knows that the diagonal of l is  1.0)
c al(1)  ar(nrows)  are not used in tridec or in trisol
        if(nrows .le. 1) return
        nrm1=nrows-1
        do 40 i=1,nrm1
        ip1=i+1
c add xi*row i    to row (i+1)   which cancels out a(i+1,i) and
c    adds xi*a(i,i+1) to a(i+1,i+1),    xi=-a(i+1,i)/a(i,i)
```

```
      xi=-al(ip1)/ad(i)
      ad(ip1)=ad(ip1)+xi*ar(i)
40    al(ip1)=-xi
c statement 40  gets l, l(i+1,i)=-xi
      return
      end
      subroutine trisol(al,ad,ar,nrows,rtside,soln)
      implicit double precision (a-h,o-z)
      dimension al(nrows), ad(nrows), ar(nrows)
      dimension  rtside(nrows), soln(nrows)
c tridiagonal backsolve, decomposed matrix from tridec is in al,ad,ar
c solving l*u*soln=rtside
c rtside is destroyed in trisol
c al(1)  ar(nrows)  are not used in tridec or in trisol
c first solve  l*(z=u*soln)  =  rtside
c then solve   u*soln = z
      if(nrows .eq. 1) soln(1)=rtside(1)/ad(1)
      if(nrows .eq. 1) return
      nrm1=nrows-1
c solve l*z=rtside, put answer in rtside
c     rtside(1)=rtside(1)
      do 20 i=2,nrows
20    rtside(i)=rtside(i)-rtside(i-1)*al(i)
c now solve  u*soln= (z=rtside)
      soln(nrows)=rtside(nrows)/ad(nrows)
      do 40 i=1,nrm1
      nmi=nrows-i
40    soln(nmi)=(rtside(nmi)-soln(nmi+1)*ar(nmi))/ad(nmi)
      return
      end
      subroutine setup
c set up difference scheme coefficients
      include 'comincl'
      kdt = 0
      t = zero
      xrt = zero
      npts = 0
      xpta(npts) = zero
      ua(npts) = zero
      x = zero
      if(initu .eq. 1) ua(npts) = truesol(x,t)
      nint = 0
      ktop = 0
      kc = 0
      do 100 ilayer = 1, nlayers
         scalea(ilayer) = solwpa(ilayer) * rhoa(ilayer) / 100.d0
         n = nintlra(ilayer)
         ktop = ktop + n + 1
         xrt = xrt + thicka(ilayer)
         dx = thicka(ilayer)/dble(n)
c        left endpoint
         kc = kc + 1
         xa(kc) = x
c        get finite difference grid in layer ilayer
         do 50 isublr = 1, n
c           right endpoint
            npts = npts + 1
            nint = nint + 1
            x = xpta(npts-1) + dx
            xpta(npts) = x
            dxa(nint) = dx
            ua(npts) = zero
            rtside(npts) = zero
            if(initu .eq. 1) ua(npts) = truesol(x,t)
            kc = kc + 1
            xa(kc) = x
50       continue
100   continue
c point indices started at 0, to get actual number
c of points add 1 to npts.
      npts = npts + 1
      celeft = scalea(1) * outrhl / 80.d0
      cert = scalea(nlayers) * outrhr / 80.d0
```

```
      nptsm1 = npts - 1
      nptsm2 = npts - 2
      nunkn = npts - 2
      if(isymrt .eq. 1) nunkn = npts - 1
      nunknt = nint - 1 + isymrt
      if(nunknt .ne. nunkn) stop'nunknt ne nunkn in setup'
      if(kc .ne. ktop) stop'kc ne ktop in setup'
c get toutfac which converts input value tf of final time in units
c specified by chartin into seconds, i.e., tfinal (sec) = tf * toutfac
      if(chartin .eq.  s') toutfac = one
      if(chartin .eq. 'i') toutfac = 60.d0
      if(chartin .eq. 'h') toutfac = 60.d0 * 60.d0
      if(chartin .eq. 'd') toutfac = 60.d0 * 60.d0 * 24.d0
      if(chartin .eq. 'w') toutfac = 60.d0 * 60.d0 * 24.d0 * 7.0d0
      if(chartin .eq. 'm') toutfac = 60.d0 * 60.d0 * 24.d0 * 30.43667d0
      if(chartin .eq. 'y') toutfac = 60.d0 * 60.d0 * 24.d0 * 365.24d0
c get tfinal (seconds) and dt (seconds)
      tfinal = tf * toutfac
      dt = tfinal / dble(ntsteps)
      tout = t / toutfac
c get finite difference scheme matrix
c********* at present assume isymrt = 0 *****************
      if(isymrt .ne. 0)stop' isymrt ne 0  have not yet programmed case'
      onema = one - alpha
      twodt = two * dt
      sixdt = 6.0d0 * dt
      threedt = 3.0d0 * dt
      i = 0
      do 300 ilayer = 1, nlayers
         n = nintlra(ilayer)
         nn = n
         if(n .lt. 2) stop'nintlra(ilayer) lt 2 in setup'
         if(ilayer .eq. nlayers) nn = n-1
         do 200 isubint = 1, nn
            i = i + 1
c
c   x(i-1)    interval i      x(i)      interval i+1        x(i+1)
c           si = scalea                 sip = scalea
c          ei = scalea*d               eip = scalea*d
c
         si = scalea(ilayer)
         sip = scalea(ilayer)
         ei = scalea(ilayer) * diffcfa(ilayer)
         eip = scalea(ilayer) * diffcfa(ilayer)
         if(isubint .eq. n) sip = scalea(ilayer+1)
         if(isubint .eq. n) eip = scalea(ilayer+1) * diffcfa(ilayer+1)
         dxi = dxa(i)
         dxip = dxa(i+1)
c        Dirichlet data at left endpoint so unknown number k is at
c        right endpoint of interval number k
         if(lumping .eq. 1) then
c  mass matrix lumped
            al(i) = -ei*alpha/dxi
            ad(i) = (si*dxi + sip*dxip)/twodt +
     &                          (ei/dxi + eip/dxip)*alpha
            ar(i) = -eip*alpha/dxip
c        for forming rtside
            aln(i) = ei*onema/dxi
            adn(i) = (si*dxi + sip*dxip)/twodt -
     &                          (ei/dxi + eip/dxip)*onema
            arn(i) = eip*onema/dxip
         else
c  mass matrix not lumped
            al(i) = -ei*alpha/dxi + si*dxi/sixdt
            ad(i) = (si*dxi + sip*dxip)/threedt +
     &                          (ei/dxi + eip/dxip)*alpha
            ar(i) = -eip*alpha/dxip + sip*dxip/sixdt
c        for forming rtside
            aln(i) = ei*onema/dxi + si*dxi/sixdt
            adn(i) = (si*dxi + sip*dxip)/threedt -
     &                          (ei/dxi + eip/dxip)*onema
            arn(i) = eip*onema/dxip + sip*dxip/sixdt
         end if
```

```
200       continue
300     continue
        if(i .ne. nunkn) stop'i ne nunkn in setup'
        call tridec(al,ad,ar,nunkn)
        return
        end
```

FORTRAN SOURCE FILE: outp.for

```
      subroutine outp(npppsi)
c do printout of results
      include 'cominc1'
      emax = zero
      write(66,600)
      write(66,600)
      write(67,600)
      write(67,600)
600   format('      ')
      dtout = dt/toutfac
      if(chartin .eq. 's') write(66,6051)  kdt, tout, dtout
      if(chartin .eq. 'i') write(66,6052)  kdt, tout, dtout
      if(chartin .eq. 'h') write(66,6053)  kdt, tout, dtout
      if(chartin .eq. 'd') write(66,6054)  kdt, tout, dtout
      if(chartin .eq. 'w') write(66,6055)  kdt, tout, dtout
      if(chartin .eq. 'm') write(66,6056)  kdt, tout, dtout
      if(chartin .eq. 'y') write(66,6057)  kdt, tout, dtout
      if(chartin .eq. 's') write(67,6051)  kdt, tout, dtout
      if(chartin .eq. 'i') write(67,6052)  kdt, tout, dtout
      if(chartin .eq. 'h') write(67,6053)  kdt, tout, dtout
      if(chartin .eq. 'd') write(67,6054)  kdt, tout, dtout
      if(chartin .eq. 'w') write(67,6055)  kdt, tout, dtout
      if(chartin .eq. 'm') write(67,6056)  kdt, tout, dtout
      if(chartin .eq. 'y') write(67,6057)  kdt, tout, dtout
6051  format(' time step # = ',i6,'      time = ',1pe11.3,
     &       ' seconds     dt = ',e11.3,' seconds')
6052  format(' time step # = ',i6,'      time = ',1pe11.3,
     &       ' minutes     dt = ',e11.3,' minutes')
6053  format(' time step # = ',i6,'      time = ',1pe11.3,
     &       ' hours     dt = ',e11.3,' hours')
6054  format(' time step # = ',i6,'      time = ',1pe11.3,
     &       ' days     dt = ',e11.3,' days')
6055  format(' time step # = ',i6,'      time = ',1pe11.3,
     &       ' weeks     dt = ',e11.3,' weeks')
6056  format(' time step # = ',i6,'      time = ',1pe11.3,
     &       ' months     dt = ',e11.3,' months')
6057  format(' time step # = ',i6,'      time = ',1pe11.3,
     &       ' years     dt = ',e11.3,' years')
c
      write(66,6061)
      write(67,6061)
6061  format(' u = scaled solution = c/scale factor,    c = ',
     &       'H2O concentration (g/cc)')
      write(66,6062)
      write(67,6062)
6062  format(' wtper = (g H2O)/(100 g dry material) = (100 c)/(rho)')
      write(66,6063)
      write(67,6063)
6063  format(' scale factor = solwpa * rho / 100 ')
c
c get concentration array ca, and weight percent array wtpera
      ipt = 0
      kc = 0
      do 99 ilayer = 1, nlayers
         n = nintlra(ilayer)
c        left endpoint
         kc = kc + 1
         ca(kc) = ua(ipt) * scalea(ilayer)
         wtpera(kc) = ca(kc) * 100.d0 / rhoa(ilayer)
         do 49 isublr = 1, n
c           right endpoint
            ipt = ipt + 1
            kc = kc + 1
            ca(kc) = ua(ipt) * scalea(ilayer)
            wtpera(kc) = ca(kc) * 100.d0 / rhoa(ilayer)
49       continue
99    continue
c now get h2olra, totalw
```

```
      half = 0.5d0
      kc = 0
      totalw = zero
      do 101 ilayer = 1, nlayers
         n = nintlra(ilayer)
         delta = thicka(ilayer)/dble(n)
         scrh = zero
c        left endpoint
         kc = kc + 1
         do 51 isublr = 1, n
c           left endpoint
            scrh = scrh + half * ca(kc)
c           right endpoint
            kc = kc + 1
            scrh = scrh + half * ca(kc)
51       continue
         h2olra(ilayer) = scrh*delta
         totalw = totalw + scrh*delta
101   continue
      itop = 0
      do 100 ilayer = 1,nlayers
         n = nintlra(ilayer)
         levx = n/npppsi
         if(levx .lt. 1) levx = 1
         ibot = itop
         itop = ibot + n
         write(66,610) (xpta(k), k=ibot,itop,levx)
610      format('   x = ',1pe14.4,4e14.4)
         write(66,611) (ua(k), k=ibot,itop,levx)
611      format('   u = ',1pe14.4,4e14.4)
         if(initu .eq. 1) then
            do 50 k = ibot, itop
               xx = xpta(k)
               scra(k+1) = ua(k) - truesol(xx,t)
               scre = abs(scra(k+1))
               if(scre .gt. emax) emax = scre
50          continue
            write(66,612) (scra(k+1), k=ibot,itop,levx)
612         format('   e = ',1pe14.4,4e14.4)
         end if
      write(66,600)
100   continue
      if(initu .eq. 1) write(66,600)
      if(initu .eq. 1) write(66,613) emax
613   format(' maximum error = ',1pe14.4)
      if(initu .eq. 1) write(66,600)
      if(initu .eq. 1) write(66,600)
c
      if(initu .eq. 0) then
      write(66,600)
      write(66,600)
      write(67,600)
      itop = 0
      do 110 ilayer = 1,nlayers
         n = nintlra(ilayer)
         levx = n/npppsi
         if(levx .lt. 1) levx = 1
         ibot = itop + 1
         itop = ibot + n
         write(66,710) (xa(k), k=ibot,itop,levx)
710      format('       x = ',1pe14.4,4e14.4)
         write(66,711) (ca(k), k=ibot,itop,levx)
711      format('       c = ',1pe14.4,4e14.4)
         write(66,712) (wtpera(k), k=ibot,itop,levx)
712      format('   wtper = ',1pe14.4,4e14.4)
      write(66,600)
110   continue
      itop0 = 0
      itop1 = 0
      write(67,6767)
6767  format(' x = location (cm)    u = scaled conc.   H2O conc.',
     &       ' (g/cc)    g H2O/100g material')
      write(67,600)
```

```
      do 120 ilayer = 1,nlayers
         n = nintlra(ilayer)
         levx = n/npppsi
         if(levx .lt. 1) levx = 1
         ibot0 = itop0
         itop0 = ibot0 + n
         ibot1 = itop1 + 1
         itop1 = ibot1 + n
         k1 = ibot1
         do 119 k=ibot0,itop0,levx
            write(67,6768) xpta(k),ua(k),ca(k1),wtpera(k1)
6768        format(1pe14.4,e20.4,e19.4,e23.4)
            k1 = k1 + levx
119      continue
120      continue
      end if
      write(66,600)
      write(66,600)
      write(67,600)
      write(67,600)
      write(66,749)
      write(67,749)
749   format(' water in each layer as grams in a 1 cm*cm cross',
     &       ' section of the material')
      write(66,750) (k, h2olra(k), k = 1,nlayers)
      write(67,750) (k, h2olra(k), k = 1,nlayers)
750   format(2(' layer = ',i3,'   H2O in layer = ',1pe11.3,'   '))
      write(66,600)
      write(66,600)
      write(67,600)
      write(66,751)
      write(67,751)
751   format(' total amount of water (grams) in a 1 cm*cm cross',
     &       ' section of the full material')
      write(66,752) totalw
      write(67,752) totalw
752   format(' total amount of H2O = ',1pe14.4)
      write(66,600)
      write(67,600)
      return
      end
```

```
GLOSSARY OF VARIABLE NAMES IN THE PROGRAM diffuse
EXPLANATION OF THE ALGORITHM USED IN THE PROGRAM diffuse


c The program DIFFUSE calculates the time history of the water
c concentration in a sample composed of distinct layers having different
c material properties (which are constant inside each layer). The
c sample is considered to be equivalent to a cylinder exposed to a
c constant relative humidity at its ends, and insulated along its
c lateral surface, with its axis along the x direction starting at x=0.
c
c As x varies from 0 to x=xrt (the right endpoint of the cylinder), one
c passes through the various layers in the sample. For a given fixed
c value of x, there is no change in the material properties or in the
c H2O concentration as y and z vary, so the mathematical model only
c involves 1 space dimension.
c
c ci ... designates that the variable being described is an input
c       variable (read in from a data file at the start of the program)
c*****  NOTE, except for  comment  which is read in with a72 format,
c*****  ALL DATA READ IN FROM THE INPUT DATA FILE, infile, IS
c*****  READ IN USING FREE FORMAT.
c co ... designates that the variable being described is an output
c       variable (whose value is computed by DIFFUSE).
c cp ... designates that the variable being described is a parameter
c       constant in the program (e.g., a maximum for the number of
c       finite difference points) which can be changed (in the include
c       file comincl) if necessary.
c cs ... designates that the variable being described is an input
c       variable which is read in from the terminal at the start
c       of the program (the names of the input and output files).
c variables without an i o p or s designator are internal variables used
c by DIFFUSE.
c
c DIFFUSE uses an include file, comincl, containing various
c        parameter, common block and declaration statements
c        needed by the routines in DIFFUSE.
c
c  CHARACTER VARIABLES
ci chartin is a character variable of length 1, specifying time units,
ci comment is a character variable of length 72, used to input and
c         output comments on a given run,
c cscr is a character scratch variable of length 3,
cs infile is a character variable of length 20 used for the input data
c        file name (infile is read in from the terminal at the start
c        of the program-follow instructions printed on screen),
cs outfile is a character variable of length 20 used for the output
c         file name (outfile is read in from the terminal at the start
c         of the program-follow instructions printed on screen),
c outcol is a character variable of length 23 used for the file name
c       of the output file containing the output data in column format,
c       outcol = <outfile with trailing blanks deleted>col,
c**** when typing in infile and outfile on the keyboard, do NOT use
c**** any blank spaces, do NOT enclose the names in apostrophes or
c**** quotes, and do not have the file names exceed the
c**** maximum file name length on the computer being used.
c**** the name of the file containing output in column format is
c**** defined to be <the output file name typed onto the screen>col
c**** which thus has 3 more characters in its name than does the
c**** regular output file.
c
c all other variables starting with the letters a-h,o-z are
c double precision,
c all other variables starting with the letters i-n are integers.
c
c******   UNITS ABBREVIATIONS
c
c g = gram
c sec = second
c cm = centimeter
c cc = cubic cm
```

```
c
c******   PHYSICAL VARIABLES
c
c xrt: the sample (axis of the cylinder) is considered to be located
c      between x=0 and x=xrt (cm).
ci nlayers: number of layers of (different) materials in the sample
c           under consideration.
cp nlmax: maximum number of layers allowed  (this is a programming
c         parameter which can be increased if necessary; its current
c         value in the include file  cominc1  is 20).
ci thicka(ilayer): array (vector) of thickness (in cm)
c                  of each layer (ilayer = 1,...,nlayers) in the sample.
ci diffcfa(ilayer): array (vector) of diffusion coefficients in
c                   each layer (ilayer = 1,...,nlayers) (units are
c                   cm*cm/sec).
ci rhoa(ilayer): is the array (vector) of density values of dry material
c                for each layer (ilayer = 1,...,nlayers)
c                (units are g/cc).
c isymrt: if 1 then the right endpoint is sealed (no moisture diffusion
c         takes place through the right endpoint of the sample). The
c         boundary condition at the right endpoint is then the no flux
c         (symmetry) boundary condition.
c         If isymrt is not 1, then the right endpoint is
c         exposed to the ambient moisture. The latter is ALWAYS the
c         the case for the current version of DIFFUSE.
ci outrhl: the relative humidity (in percent) outside of the
c          left endpoint (x=0) of the sample.
ci outrhr: the relative humidity (in percent) outside of the
c          right endpoint (x=xrt) of the sample.
c cross section: for purposes of 1-dimensional diffusion calculations
c                we consider the situation of a 1 square cm cross
c                section of material, so a length of 1 cm corresponds to
c                a volume of 1 cc (cubic cm). The numerical solution
c                procedure is a mass conservative finite difference
c                method. When alpha = 1. and lumping = 1 (see below):
c                this method satisfies a maximum
c                principle (the range of the solution values
c                must stay within that of the initial and boundary
c                data), and it is equivalent to the
c                lumped fully implicit finite element method
c                with linear elements for this diffusion problem.
c concentration: denoted by c or c(x) or c(x,t), is the concentration
c                of H2O at a particular location x in the sample,
c                at some time t (in units of grams of H2O/cc, or
c                grams of H2O/100 grams of dry material).
c                The solution of the diffusion equation is done in
c                terms of grams of H2O/cc, and UNLESS OTHERWISE NOTED,
c                ALL CONCENTRATIONS ARE IN GRAMS OF H2O/cc.
c celeft: equilibrium concentration of H2O in the material at the left
c         endpoint of the sample when maintained at the relative
c         humidity value outrhl.
c cert: equilibrium concentration of H2O in the material at the right
c       endpoint of the sample when maintained at the relative
c       humidity value outrhr.
c scalea(ilayer): concentration scale factor array (vector)
c                 (ilayer = 1,...,nlayers). If the material in layer
c                 ilayer and some "standard" material are both
c                 maintained to equilibrium at some given relative
c                 humidity, then the H2O concentration in the material
c                 in layer ilayer will equal scalea(ilayer) * the H2O
c                 concentration in the "standard" material. Thus at
c                 equilibrium the normalized or scaled concentrations
c                       c(ilayer)/scalea(ilayer)
c                 will all be the same. It is being ASSUMED that right
c                 at the interface I between 2 materials;
c
c                    material i       |     material i+1
c                 --------------------|---------------------
c                                     I
c
c                 the relationship  c(i)/scalea(i) = c(i+1)/scalea(i+1)
c                 always holds (even before equilibrium is attained).
c                 scalea(ilayer) is defined as a dimensionless constant.
```

```
ci rhsol  relative humidity (%) at which equilibrium moisture
c         solubility data (solwpa) for all the materials have been
c         measured.
ci solwpa(ilayer): is the array (vector) of the equilibrium solubility
c                  of H2O in the material of layer ilayer
c                  (ilayer= 1,...,nlayers) maintained at rhsol% relative
c                  humidity. units are g H2O / 100 g dry material =
c                  weight percent of H2O.
c***********       NOTE immediately after solwpa has been read in and
c                  then printed out, it is multiplied by
c                           80. / rhsol
c                  to rescale it to correspond to saturation levels
c                  at 80% relative humidity (since that is what
c                  was assumed for the original version of this code).
c                  NOTE scalea(ilayer) is set
c                  = (solwpa(ilayer) * rhoa(ilayer) / 100.) / (1 g/cc)
c                  Note also celeft = (scalea(1) * outrhl / 80.)*1 g/cc
c                  and   cert = (scalea(nlayers) * outrhr / 80.)*1 g/cc
c
c******   NUMERICAL SOLUTION PROCEDURE VARIABLES / OUTPUT VARIABLES
c
ci nintlra(ilayer): array (vector) of number of subintervals each layer
c                   (ilayer = 1,...,nlayers) is subdivided into for the
c                   finite difference method. Thus the thickness of each
c                   subinterval in layer ilayer is
c                   thicka(ilayer) / nintlra(ilayer).
c****               the value of nintlra(ilayer) MUST be at least 2
c****               for each material layer.
c nint: total number of subintervals into which [0, xrt] is
c       subdivided for the numerical solution procedure. nint is the sum
c       over ilayer = 1,...,nlayers of nintlra(ilayer).
c npts: is the number of computational points where c(x) is calculated,
c       including the two boundary points where the value of c(x) is
c       specified; npts = nint + 1
cp nptsmax: maximum number of computational points allowed (this is a
c           programming parameter which can be increased if necessary;
c           its current value in the include file  cominc1  is 800).
c nptsm1: is npts - 1
c nptsm2: is npts - 2
c nunkn: is the number of computational points where the unknown value
c        of c(x) is calculated at each time step = npts - 2 = all the
c        grid points - the two endpoints where (known) boundary data
c        for c is specified. (When isymrt = 1, nunkn = npts - 1)
c xpta(ipt): array (vector) of x point values at which c(x) is
c            calculated (ipt = 0,...,nptsm1) (units for xpta is cm).
c            The two endpoints (x=0 and x=xrt) are included in xpta.
c dxa(int): is the mesh spacing array (interval size array);
c           dxa(int) = xpta(int)-xpta(int-1)  (for int = 1,...,nint).
co ua(ipt): is the array (vector) of the approximate values at a given
c           time t for c(xpta(ipt),t)/scalea(ilayer) where ilayer is the
c           number of the material layer containing xpta(ipt)
c           (ipt = 0,...,npts-1). Note from the discussion of scalea,
c           there is no ambiguity in the value of ua at the interface
c           between two layers of material (which is the rationale for
c           the definition of u = c/scalea). THE NUMERICAL DISCRETIZATION
c           (APPROXIMATION) OF THE DIFFUSION PROBLEM IS FORMULATED AND
c           SOLVED IN TERMS OF ua.
c           Note the boundary conditions for u are (cf. the specific
c           definition of scalea) u = outrhl / 80. at x=0,
c           and u = outrhr / 80. at x=xrt.
co ca(k), xa, ktop: ca(k) is an array (vector) of H2O concentration
c                   values in the sample (k = 1,...,ktop). The x value
c                   corresponding to ca(k) is xa(k). The values of xa
c                   range over the values of xpta. At a point
c                   xa(k) = xpta(ipt) which is not an interface between
c                   two layers,
c                   ca(k) = ua(ipt) * scalea(ilayer)
c                   where ilayer is the index of the layer containing
c                   xpta(ipt). AT INTERFACE POINTS BETWEEN TWO MATERIAL
c                   LAYERS THERE ARE 2 VALUES OF ca GIVEN;
c             ca(k) = ua * (scalea value from left side of interface)
c           ca(k+1) = ua * (scalea value from right side of interface)
c           in which case xa(k) = xa(k+1) = the interface location.
```

```
cp kmax: maximum value of ktop allowed  (this is a programming
c        parameter which can be increased if necessary--it is currently
c        set to nptsmax+nlmax in comincl which will always be adequately
c        large).
co wtpera(k): is the array (vector) of concentration values in units of
c             g H2O / 100 g dry material   (weight percent of H2O)
c             for k = 1,...,ktop. wtpera(k) =
c             (ca(k) / rhoa(ilayer corresponding to xa(k))) * 100.
c             At a pair of indices k, k+1 corresponding to an interface,
c             rhoa in the denominator takes the value from the left,
c             then the right side of the interface at k, k+1
c             respectively.
c e.g., a concentration weight percent of .6 means .6 g H2O/100 g dry
c material.
co h2olra(ilayer): array (vector) of total amount of H2O (grams) in
c                  each layer (ilayer = 1,...,nlayers). This is
c                  obtained from ca and xa (trapezoidal quadrature).
c                  The cross section size is 1 square cm.
co totalw: is the total amount of water in the sample (g H2O) = the sum
c          over ilayer = 1,...,nlayers of h2olra(ilayer).
c          The cross section size is 1 square cm.
c t: is the current time value at which the approximate solution of the
c    diffusion problem is being obtained (units are seconds).
c tfinal: final time value for which the approximate solution is to be
c         obtained (units are seconds).
ci chartin: character variable of length 1 which specifies the units for
c           the input value of the final time, tf, and the units
c           for values of time, tout, in the output file.
c           chartin = 's' for seconds, 'i' for minutes, 'h' for hours,
c           'd' for days, 'w' for weeks,
c           'm' for months (=30.43667 days),
c           or 'y' for years (=365.24 days).
ci tf: input value of the final time at which the concentration in the
c      sample is to be obtained. (Units for tf and tout are specified
c      by chartin.)
co tout: variable containing current output time value.
c toutfac: tout = t / toutfac (convert time in sec to output time
c          units specified by chartin).
ci ntsteps: total number of time steps to be used to obtain the
c           approximate solution of the diffusion equation up to
c           t = tfinal (time steps are kdt = 1,2,...,ntsteps).
c kdt: time step number (kdt = 1,...,ntsteps) (kdt = 0 during
c      input of data and initialization of the numerical solution
c      procedure).
c dt: time step size (seconds) = tfinal / ntsteps
ci levt: print out approximate solution values every  levt  time steps,
c        i.e., at t=0 (initial data), t = levt*dt, t = 2 * levt*dt, etc.
ci initu: initialization indicator, set = 0 to set ua=0 at t=0; set = 1
c         to use the function truesol to initialize ua (for use in
c         testing the program with a known analytical solution function)
ci alpha: time averaging parameter; alpha = 1. for pure implicit,
c         alpha = .5 for Crank-Nicolson (Crank-Nicolson is second order
c         accurate but theoretically not as stable as pure implicit).
c***      for this application it is probably best to choose alpha=.5
ci lumping: if lumping = 1 then "lump the mass matrix," i.e., do not
c           use the two grid points adjacent to the center point to
c           average the time differencing -- when alpha = 1., lumping=1
c           enforces the discrete maximum principle. When lumping is
c           not equal 1 the resulting difference scheme is the
c           full finite element method with linear elements.
c***        for this application it is probably best to choose lumping=0
c
c
c***********      COMMENTS ON CHOOSING SPATIAL MESH AND TIME STEP SIZE
c
c In order to verify that sufficiently many finite difference
c subintervals (nintlra) have been used for each material
c layer, and that a sufficiently small
c time step size has been chosen (ntsteps large enough) in order to
c obtain an accurate approximate solution to the moisture diffusion
c problem using DIFFUSE, it is suggested that one do the following.
c (1) do a printout of the numerical solution values at an "early"
c time (well before the first time value one is specifically
```

```
c interested in), and make sure there are no unnatural oscillations
c in the solution values -- if there are then increase ntsteps.
c (2) double ntsteps and double each nintlra(ilayer) value and check
c to see if the changes in the numerical solution values at the times
c and spatial locations in which one is interested have changed
c by more than an acceptable amount. Keep increasing the resolution
c of the spatial grid and time stepping until the numerical results
c vary by less than whatever tolerance is desired.
c
c Note that one must have a separate "numerical scheme layer" for each
c distinct material layer. However one is free to have adjacent
c specified material layers whose physical properties are identical.
c For example in a sample with same material from x=2 to x=4,
c one is free to use one layer from x=2 to x=4, or one could use
c two layers (x=2 to 3, and x=3 to 4) or
c one could take x=2 to 2.5 and x=2.5 to 4 etc. One can use
c this to, e.g., "grade the numerical mesh" within one material, or vary
c the amount of printout in one layer without changing the Fortran.
c
ci comment: up to five lines of comments (72 characters each) at the
c           end of the input file (after the line where the value of
c           the variable  lumping  is specified) are read into the
c           character variable comment (with a72 format)
c           and are printed out at the beginning of the output file.
c npppsi: governs the number of points printed (in the output file)
c         for each material layer. At present npppsi is 4 which means
c         that, as long as npppsi evenly divides nintlra(ilayer),
c         there will be 5 output points from material layer
c         number ilayer (the 2 endpoints and 3 evenly spaced interior
c         points). the value of npppsi is set at the beginning of
c         the main program DIFFUSE.
c kleftbc, krtbc: indicate type of boundary data at the left, right
c                 endpoints -- not used within the current version
c                 of DIFFUSE.
c al, ad, ar, rtside: vectors used in the finite difference method
c                     solution procedure for evaluating ua at each
c                     time step.
c aln, adn, arn: vectors used in forming rtside.
c scra: is a scratch array.
c zero: the constant 0.
c one: the constant 1.
c two: the constant 2.
c
c
c******  FORMULATION OF THE DIFFUSION PROBLEM IN TERMS OF u = c / scalea
c
c The flux F at a given location is -d(x)c'(x) where c' denotes the
c partial derivative of c with respect to x. At an interface point
c the limits of this expression as x approaches the interface from
c either side must be the same. Setting e(x) = d(x)*scalea(ilayer)
c where ilayer is the layer containing the point x, the flux F
c is then -d(x)*scalea * c'(x)/scalea = -e(x)u'(x), and as described
c above, u(x) is continuous across the material interfaces.
c We now consider a mass balance finite difference equation at
c the finite difference grid point x(i) in the following diagram:
c
c
c       u(i-1)      F(i)       u(i)          F(i+1)         u(i+1)
c         o          |          o              |             o
c       x(i-1)   x(i-1/2)     x(i)         x(i+1/2)        x(i+1)
c                  e(i)                     e(i+1)
c                scalea(i)                 scalea(i+1)
c
c Note in the diagram we are using the notation scalea(i) = the value
c of scalea(ilayer) for the material layer containing x(i-1/2).
c
c The net amount of H2O passing into the interval [x(i-1/2), x(i+1/2)]
c during an interval of time dt is  dt*(-F(i+1)+F(i)) = (approximately)
c        dt * e(i+1) * [(u(i+1) - u(i))/(x(i+1) - x(i))]
c      - dt * e(i) * [(u(i) - u(i-1))/(x(i) - x(i-1))]
c (To make the units come out correctly, one would multiply by the
c 1 square cm cross section of the cylinder.)
c
```

```
c By conservation of mass, this is balanced by the change deltau (during
c the time dt) in the scaled concentration u in the interval
c [x(i-1/2), x(i+1/2)] (i.e., in the volume formed by sweeping out the
c 1 square cm cross section over this interval). The change deltau
c results in the (approximate) change of H2O content in the interval
c given by
c          .5*(x(i+1)-x(i)) * scalea(i+1) * deltau
c        + .5*(x(i)-x(i-1)) * scalea(i) * deltau
c
c Setting the net amount of H2O passing into the interval (13-14 lines
c above) equal to the expression immediately above gives the finite
c difference equation at the (interior) grid point x(i). Values for
c u at the two endpoints (x=0, xrt) are specified (determined by the
c ambient relative humidity). THESE EQUATIONS ARE IDENTICAL TO THE
c RESULT OF APPLYING THE LUMPED FINITE ELEMENT METHOD WITH LINEAR
c ELEMENTS TO THE EQUATION:
c
c          scalea (du/dt) = (e u')'
c or
c          scalea (du/dt) = (scalea*d u')'
c or
c          (solwpa*rhoa/100.) (du/dt) = ((solwpa*rhoa/100.)*d u')'
c
c where du/dt designates the partial derivative of u with respect to t.
c Note the boundary conditions for u at the two endpoints are given by
c          u = outrhl / 80. at x = 0, and
c          u = outrhr / 80. at x = xrt.
c The concentration of H2O is then given by  c = scalea * u.
c
c The finite difference equations above result in a tridiagonal
c linear system of equations which needs to be solved whenever
c u (and thus c) are obtained at a time t+dt using the current (known)
c values at time t and the boundary data.
c
c******   SUBROUTINES IN diffuse
c
c diffuse: main program
c initl: reads in data from input file (physical constants and
c        specifications for the finite difference method).
c setup: sets up difference scheme coefficients.
c advance: advances the approximate solution one dt time step.
c outp: writes specified output to output file.
c tridec: routine used to decompose (factor) the
c         tridiagonal linear system (matrix) from the
c         finite difference equations for advancing the approximate
c         solution of the diffusion problem one time step.
c trisol: routine used at each time step to solve (backsolve) the
c         factored tridiagonal linear system of finite difference
c         equations in order to get the approximate solution
c         values at the next time level.
c truesol, source: functions used to test the program
c                  using a known exact solution.
c dirl, dirr: functions used to get the boundary values (at x=0, x=xrt)
c             for the scaled solution u.
c             The Fortran functions dirl and dirr are functions of
c             time. In the present version of DIFFUSE they only depend
c             on the constants outrhl, outrhr (their argument, i.e.,
c             time, is not used). The structure of DIFFUSE is such that
c             time dependent boundary data could be programmed into
c             dirl and dirr. Note some compilers may issue a message
c             that the argument (time) of dirl and dirr is not used.
c             In this case, such a message is of no consequence.
c parameter specifications, variable declarations and common blocks
c are in the file  comincl  which is accessed by include statements.
c**** the file comincl should reside in the same directory which
c**** contains the Fortran program routines of DIFFUSE.
c
```

```
c***  SAMPLE INPUT DATA (schematic sample)
c***  this is not "real" physical data, but serves to explain the
c***  layout of the input data. See actual sample input data and
c***  corresponding output from running the program on a
c***  real application.
c***  SAMPLE INPUT DATA  (free format except for a72 format for comment)
c***  sample input data starts on line below.
3 nlayers # material layers; for each layer, read in its physical props.
40. rhsol  rel humidity at which solwpa data has been measured (%)
1.    thicka: thickness of this layer in centimeters (cm)
1.    diffcfa: diffusion coeff. of material in this layer (cm*cm/sec)
1.    rhoa: density of the (dry) material in this layer (g/cc)
1.    solwpa: equil. sol. of H2O in material in this layer (wt %) @rhsol
 2.  thicka: thickness of this layer in cm
 2.  diffcfa: diffusion coeff. of material in this layer (cm*cm/sec)
 2.  rhoa: density of the (dry) material in this layer (g/cc)
 2.  solwpa: equil. sol. of H2O in material in this layer (wt %) @rhsol
  3.  thicka: thickness of this layer in cm
  3.  diffcfa: diffusion coeff. of material in this layer (cm*cm/sec)
  3.  rhoa: density of the (dry) material in this layer (g/cc)
  3.  solwpa: equil. sol. of H2O in material in this layer (wt %) @rhsol
45. 55. outrhl, outrhr: rel. hum. (%) outside of the left, rt  endpt
10  nintlra: no. of finite diff. subintervals 1# for each layer(layer 1)
20  nintlra: no. of finite diff. subintervals 1# for each layer(layer 2)
30  nintlra: no. of finite diff. subintervals 1# for each layer(layer 3)
 'h'  chartin: character variable of length 1 giving input time units
8. tf: final time (units specified by chartin) calc. carried out to
400  ntsteps: number of finite difference method time steps to get to tf
20  levt: print out approx. solution values every levt time steps
0  initu: if 0, start with 0 concentration, if 1 test with known c(x,t)
0.5 alpha: time ave. parameter (.5 is Crank-Nicolson=recommended choice)
0 lumping: 0 means use the full finite element method=recommended choice
 comment1: any remaining lines (up to a maximum of 5) in the input
 comment2: file are considered to be comments which are printed at the
 comment3: beginning of the output file
 comment4: the comment lines (if any) are read in with a72 format so
 comment5: no quote marks are to be used to delimit the comment text
```

PROGRAM INSTRUCTIONS:

The include file COMINCL should be located in the same directory as the Fortran source files when compilation is done. When run, DIFFUSE will ask the user to type in the input file name (no blanks, do not enclose in quotes), and then the output file name, e.g., outfile. DIFFUSE also creates a column formatted output file , for instance, outfilecol (name = output file name with col appended).

INPUT INSTRUCTIONS:

A schematic sample input file is given at the end of the glossary on page A-22.

All values read in from the input data file are read with free format (blank space(s) act as delimiters, and comments may follow after a blank space after the last data entry on a line of input), except that at the end of the input file (after the integer lumping has been read in), from 0 to 5 lines of comments are input with a72 format (and then put into the output file.

The first item of input is the number of material layers (nlayers) in the sample being considered. The relative humidity (rhsol, in %) at which the input equilibrium H2O solubility data was measured is input on the next line. There are then four lines of input for each material layer (running from left to right across the sample). These lines contain the thickness (thicka, in cm) of the layer, the diffusion coefficient of the material in the layer (diffcfa, cm^2/s), the density (rhoa, g/cc) of the material in the layer, and the equilibrium solubility (solwpa) of H2O in the material of the layer when maintained at rhsol relative humidity (solwpa is weight % = g of H2O/100 g dry material).

Following this input data for each of the nlayers material layers, the values of the relative humidity maintained at the left and right endpoints is supplied on one line (outrhl, outrhr, %). Then for each of the material layers there is one input line containing the number of finite difference (element) subintervals (nintlra) into which the layer is subdivided for the numerical solution procedure (see comments in the Glossary regarding suggested procedures for determining an appropriate choice of nintlra, and of the time step dt for the algorithm).

The next line of input (chartin, one letter within quotes, e.g., 'y') specifies the time units for the value of the final time value up to which the time history of the moisture concentration in the material is calculated. The character variable chartin is also used to specify units of time in the output (chartin = 's' for seconds, 'i' for minutes, 'h' for hours, 'd' for days, 'w' for weeks, 'm' for months, and 'y' for years). The final time (tf, in units specified by chartin) is given on the next line of input, followed by the number of finite difference time steps (ntsteps) to be used to get to tf (so the finite difference time step deltat = tf/ntsteps).

The next line contains levt which governs the frequency of printing out the concentration profiles (e.g., levt=100 means there is a printout every 100 time steps of the solution procedure, i.e., at time intervals of levt*deltat).

The next line contains initu which is 0 to run the program on a sample starting with zero moisture content (initu is 1 to test run the program with a known solution of the diffusion equation).
The next line contains the time step parameter choice (alpha = .5 is Crank-Nicolson which is recommended for this application, alpha = 1 is pure implicit time stepping).

The next line contains the indicator governing whether the mass matrix in the finite element method will not be lumped (lumping = 0 which is recommended), or will be lumped (lumping = 1). If there are additional lines in the input file they (up to five lines) will be read in with a72 format and then reproduced in the output file.

Example No. 1: Run with F4/C20/S4/C20/S4/C20/F4 dt = 1 months
F = 3113 glass-epoxy
C = blasa wood
S = G10 glass epoxy (without copper screen)

FORTRAN SOURCE FILE: inp5

```
  7 nlayers   next, for each layer, read in its physical properties
80.  rhsol (%): rel hum at which equil H2O solubilities solwpa were measured
.2820    thicka: thickness of this layer in centimeters (cm)
5.81d-10   diffcfa: diffusion coeff. of material in this layer (cm*cm/sec)
1.765   rhoa: density of the (dry) material in this layer (g/cc)
.664   solwpa: equil. sol. of H2O in material in this layer (wt %) @rhsol%RH
 4.496 thicka: thickness of this layer in cm
 6.59d-6 diffcfa: diffusion coeff. of material in this layer (cm*cm/sec)
 .091 rhoa: density of the (dry) material in this layer (g/cc)
 14.361 solwpa: equil. sol. of H2O in material in this layer (wt %) @rhsol%RH
  .096 thicka: thickness of this layer in cm
  5.05d-10 diffcfa: diffusion coeff. of material in this layer (cm*cm/sec)
  1.827 rhoa: density of the (dry) material in this layer (g/cc)
  .72 solwpa: equil. sol. of H2O in material in this layer (wt %) @rhsol%RH
 4.064 thicka: thickness of this layer in cm
 6.59d-6 diffcfa: diffusion coeff. of material in this layer (cm*cm/sec)
 .091 rhoa: density of the (dry) material in this layer (g/cc)
 14.361 solwpa: equil. sol. of H2O in material in this layer (wt %) @rhsol%RH
  .096 thicka: thickness of this layer in cm
  5.05d-10 diffcfa: diffusion coeff. of material in this layer (cm*cm/sec)
  1.827 rhoa: density of the (dry) material in this layer (g/cc)
  .72 solwpa: equil. sol. of H2O in material in this layer (wt %) @rhsol%RH
 4.496 thicka: thickness of this layer in cm
 6.59d-6 diffcfa: diffusion coeff. of material in this layer (cm*cm/sec)
 .091 rhoa: density of the (dry) material in this layer (g/cc)
 14.361 solwpa: equil. sol. of H2O in material in this layer (wt %) @rhsol%RH
.2820    thicka: thickness of this layer in centimeters (cm)
5.81d-10   diffcfa: diffusion coeff. of material in this layer (cm*cm/sec)
1.765   rhoa: density of the (dry) material in this layer (g/cc)
.664   solwpa: equil. sol. of H2O in material in this layer (wt %) @rhsol%RH
80.  80.  outrhl, outrhr: rel. humidity(%) outside of the sample at left, rt end
  4  nintlra: no. of finite diff. subintervals 1# for each layer(layer 1)
 20  nintlra: no. of finite diff. subintervals 1# for each layer(layer 2)
  4  nintlra: no. of finite diff. subintervals 1# for each layer(layer 3)
 20  nintlra: no. of finite diff. subintervals 1# for each layer(layer 4)
  4  nintlra: no. of finite diff. subintervals 1# for each layer(layer 5)
 20  nintlra: no. of finite diff. subintervals 1# for each layer(layer 6)
  4  nintlra: no. of finite diff. subintervals 1# for each layer(layer 7)
 'm'  chartin: character variable of length 1 giving input time units
1200. tf: final time (units specified by chartin) calculation carried out to
1200  ntsteps: number of finite difference method time steps to get to tf
240  levt: print out approx. solution values every levt time steps
0  initu: if 0, start with 0 concentration, if 1 test with known c(x,t)
0.5 alpha: sets time ave., use .5 (Crank-Nicolson)=best choice; or 1. (implicit)
0 lumping: if 0 use full finite element method-best; if 1 lump the mass matrix
 face  core/balsa  Cu-sheet  core/balsa Cu-sheet core/balsa face
 center material (core-balsa) has symmetry pt at its midpt. This is instead
    of treating it as two = size materials
    (with the same material properties and 1/2 the thickness) & with the
    symmetry pt between them. (numerical solution is unchanged) 3/11/95
```

Example No. 1: Run with F4/C20/S4/C20/S4/C20/F4 dt = 1 months
F = 3113 glass-epoxy
C = blasa wood
S = G10 glass epoxy (without copper screen)

FORTRAN OUTPUT FILE: out5

```
 output file name is out5
 input file name is inp5
 number of material layers is          7
 rhsol = relative humidity in % at which the
 equilibrium solubility (wt %) of H2O in all the
material layers was measured; rhsol (%) =  80.00
         layer number          1
 2.820000000000000E-001  thicka (cm)
 5.810000000000000E-010  diffcfa (cm*cm/sec)
      1.765000000000000  rhoa (g/cc)
 6.640000000000000E-001  solwpa(wt% @rhsol%RH gH2O/100g dry m)
         layer number          2
      4.496000000000000  thicka (cm)
 6.590000000000000E-006  diffcfa (cm*cm/sec)
 9.100000000000000E-002  rhoa (g/cc)
     14.361000000000000  solwpa(wt% @rhsol%RH gH2O/100g dry m)
         layer number          3
 9.600000000000000E-002  thicka (cm)
 5.050000000000000E-010  diffcfa (cm*cm/sec)
      1.827000000000000  rhoa (g/cc)
 7.200000000000000E-001  solwpa(wt% @rhsol%RH gH2O/100g dry m)
         layer number          4
      4.064000000000000  thicka (cm)
 6.590000000000000E-006  diffcfa (cm*cm/sec)
 9.100000000000000E-002  rhoa (g/cc)
     14.361000000000000  solwpa(wt% @rhsol%RH gH2O/100g dry m)
         layer number          5
 9.600000000000000E-002  thicka (cm)
 5.050000000000000E-010  diffcfa (cm*cm/sec)
      1.827000000000000  rhoa (g/cc)
 7.200000000000000E-001  solwpa(wt% @rhsol%RH gH2O/100g dry m)
         layer number          6
      4.496000000000000  thicka (cm)
 6.590000000000000E-006  diffcfa (cm*cm/sec)
 9.100000000000000E-002  rhoa (g/cc)
     14.361000000000000  solwpa(wt% @rhsol%RH gH2O/100g dry m)
         layer number          7
 2.820000000000000E-001  thicka (cm)
 5.810000000000000E-010  diffcfa (cm*cm/sec)
      1.765000000000000  rhoa (g/cc)
 6.640000000000000E-001  solwpa(wt% @rhsol%RH gH2O/100g dry m)
         0  if 0 (1) right end is exposed (sealed)
     80.000000000000000       80.000000000000000  outside relative humidity (%
)
 at the left, right boundary (end of the material)
         4  # finite diff subintervals in layer          1
        20  # finite diff subintervals in layer          2
         4  # finite diff subintervals in layer          3
        20  # finite diff subintervals in layer          4
         4  # finite diff subintervals in layer          5
        20  # finite diff subintervals in layer          6
         4  # finite diff subintervals in layer          7
 m  char var of length 1 --- input time units
 time units are: s = seconds, i = minutes,
 time units are: h = hours, d = days, w = weeks,
 time units are: m = months (30.43667 days/month)
 time units are: y = years (365.24 days/year)
   1200.000000000000000  final time (in given units) calc carried out to
      1200  # of finite diff time steps to get to tf
       240 print approx soln values every levt time steps
         0  if 0 (1) start with 0 (known test) concen.
 5.000000000000000E-001  time ave parameter, 1. = implicit, .5 = C.N.
         0  if 1 lump mass matrix, else do not lump
```

face core/balsa Cu-sheet core/balsa Cu-sheet core/balsa face
center material (core-balsa) has symmetry pt at its midpt. This is instead
of treating it as two = size materials
(with the same material properties and 1/2 the thickness) & with the
symmetry pt between them. (numerical solution is unchanged) 3/11/95

time step # = 0 time = 0.000E+00 months dt = 1.000E+00 months
u = scaled solution = c/scale factor, c = H2O concentration (g/cc)
wtper = (g H2O)/(100 g dry material) = (100 c)/(rho)
scale factor = solwpa * rho / 100

x =	0.0000E+00	7.0500E-02	1.4100E-01	2.1150E-01	2.8200E-01
u =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
x =	2.8200E-01	1.4060E+00	2.5300E+00	3.6540E+00	4.7780E+00
u =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
x =	4.7780E+00	4.8020E+00	4.8260E+00	4.8500E+00	4.8740E+00
u =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
x =	4.8740E+00	5.8900E+00	6.9060E+00	7.9220E+00	8.9380E+00
u =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
x =	8.9380E+00	8.9620E+00	8.9860E+00	9.0100E+00	9.0340E+00
u =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
x =	9.0340E+00	1.0158E+01	1.1282E+01	1.2406E+01	1.3530E+01
u =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
x =	1.3530E+01	1.3601E+01	1.3671E+01	1.3742E+01	1.3812E+01
u =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00

x =	0.0000E+00	7.0500E-02	1.4100E-01	2.1150E-01	2.8200E-01
c =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
wtper =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
x =	2.8200E-01	1.4060E+00	2.5300E+00	3.6540E+00	4.7780E+00
c =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
wtper =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
x =	4.7780E+00	4.8020E+00	4.8260E+00	4.8500E+00	4.8740E+00
c =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
wtper =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
x =	4.8740E+00	5.8900E+00	6.9060E+00	7.9220E+00	8.9380E+00
c =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
wtper =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
x =	8.9380E+00	8.9620E+00	8.9860E+00	9.0100E+00	9.0340E+00
c =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
wtper =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
x =	9.0340E+00	1.0158E+01	1.1282E+01	1.2406E+01	1.3530E+01
c =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
wtper =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
x =	1.3530E+01	1.3601E+01	1.3671E+01	1.3742E+01	1.3812E+01
c =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
wtper =	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00

water in each layer as grams in a 1 cm*cm cross section of the material
layer = 1 H2O in layer = 0.000E+00 layer = 2 H2O in layer = 0.000E+00
layer = 3 H2O in layer = 0.000E+00 layer = 4 H2O in layer = 0.000E+00
layer = 5 H2O in layer = 0.000E+00 layer = 6 H2O in layer = 0.000E+00
layer = 7 H2O in layer = 0.000E+00 layer =

total amount of water (grams) in a 1 cm*cm cross section of the full material
total amount of H2O = 0.0000E+00

```
time step # =    240      time =   2.400E+02 months     dt =   1.000E+00 months
u = scaled solution = c/scale factor,    c = H2O concentration (g/cc)
wtper = (g H2O)/(100 g dry material) = (100 c)/(rho)
scale factor = solwpa * rho / 100
  x =     0.0000E+00   7.0500E-02   1.4100E-01   2.1150E-01   2.8200E-01
  u =     1.0000E+00   7.9333E-01   5.8717E-01   3.8200E-01   1.7832E-01

  x =     2.8200E-01   1.4060E+00   2.5300E+00   3.6540E+00   4.7780E+00
  u =     1.7832E-01   1.7808E-01   1.7789E-01   1.7775E-01   1.7765E-01

  x =     4.7780E+00   4.8020E+00   4.8260E+00   4.8500E+00   4.8740E+00
  u =     1.7765E-01   1.5610E-01   1.3482E-01   1.1379E-01   9.3006E-02

  x =     4.8740E+00   5.8900E+00   6.9060E+00   7.9220E+00   8.9380E+00
  u =     9.3006E-02   9.2956E-02   9.2939E-02   9.2956E-02   9.3006E-02

  x =     8.9380E+00   8.9620E+00   8.9860E+00   9.0100E+00   9.0340E+00
  u =     9.3006E-02   1.1379E-01   1.3482E-01   1.5610E-01   1.7765E-01

  x =     9.0340E+00   1.0158E+01   1.1282E+01   1.2406E+01   1.3530E+01
  u =     1.7765E-01   1.7775E-01   1.7789E-01   1.7808E-01   1.7832E-01

  x =     1.3530E+01   1.3601E+01   1.3671E+01   1.3742E+01   1.3812E+01
  u =     1.7832E-01   3.8200E-01   5.8717E-01   7.9333E-01   1.0000E+00


    x =     0.0000E+00   7.0500E-02   1.4100E-01   2.1150E-01   2.8200E-01
    c =     1.1720E-02   9.2976E-03   6.8814E-03   4.4768E-03   2.0898E-03
wtper =     6.6400E-01   5.2677E-01   3.8988E-01   2.5365E-01   1.1840E-01

    x =     2.8200E-01   1.4060E+00   2.5300E+00   3.6540E+00   4.7780E+00
    c =     2.3303E-03   2.3273E-03   2.3248E-03   2.3229E-03   2.3216E-03
wtper =     2.5608E+00   2.5574E+00   2.5547E+00   2.5526E+00   2.5512E+00

    x =     4.7780E+00   4.8020E+00   4.8260E+00   4.8500E+00   4.8740E+00
    c =     2.3368E-03   2.0534E-03   1.7734E-03   1.4968E-03   1.2234E-03
wtper =     1.2791E-01   1.1239E-01   9.7068E-02   8.1926E-02   6.6964E-02

    x =     4.8740E+00   5.8900E+00   6.9060E+00   7.9220E+00   8.9380E+00
    c =     1.2155E-03   1.2148E-03   1.2146E-03   1.2148E-03   1.2155E-03
wtper =     1.3357E+00   1.3349E+00   1.3347E+00   1.3349E+00   1.3357E+00

    x =     8.9380E+00   8.9620E+00   8.9860E+00   9.0100E+00   9.0340E+00
    c =     1.2234E-03   1.4968E-03   1.7734E-03   2.0534E-03   2.3368E-03
wtper =     6.6964E-02   8.1926E-02   9.7068E-02   1.1239E-01   1.2791E-01

    x =     9.0340E+00   1.0158E+01   1.1282E+01   1.2406E+01   1.3530E+01
    c =     2.3216E-03   2.3229E-03   2.3248E-03   2.3273E-03   2.3303E-03
wtper =     2.5512E+00   2.5526E+00   2.5547E+00   2.5574E+00   2.5608E+00

    x =     1.3530E+01   1.3601E+01   1.3671E+01   1.3742E+01   1.3812E+01
    c =     2.0898E-03   4.4768E-03   6.8814E-03   9.2976E-03   1.1720E-02
wtper =     1.1840E-01   2.5365E-01   3.8988E-01   5.2677E-01   6.6400E-01


water in each layer as grams in a 1 cm*cm cross section of the material
layer =   1   H2O in layer =   1.943E-03    layer =   2   H2O in layer =   1.045E-02
layer =   3   H2O in layer =   1.705E-04    layer =   4   H2O in layer =   4.937E-03
layer =   5   H2O in layer =   1.705E-04    layer =   6   H2O in layer =   1.045E-02
layer =   7   H2O in layer =   1.943E-03    layer =

total amount of water (grams) in a 1 cm*cm cross section of the full material
total amount of H2O =     3.0072E-02
```

```
time step # =    480      time =   4.800E+02 months     dt =   1.000E+00 months
 u = scaled solution = c/scale factor,    c = H2O concentration (g/cc)
 wtper = (g H2O)/(100 g dry material) = (100 c)/(rho)
 scale factor = solwpa * rho / 100
   x =     0.0000E+00    7.0500E-02    1.4100E-01    2.1150E-01    2.8200E-01
   u =     1.0000E+00    8.2597E-01    6.5234E-01    4.7950E-01    3.0785E-01

   x =     2.8200E-01    1.4060E+00    2.5300E+00    3.6540E+00    4.7780E+00
   u =     3.0785E-01    3.0766E-01    3.0749E-01    3.0737E-01    3.0728E-01

   x =     4.7780E+00    4.8020E+00    4.8260E+00    4.8500E+00    4.8740E+00
   u =     3.0728E-01    2.8703E-01    2.6699E-01    2.4718E-01    2.2760E-01

   x =     4.8740E+00    5.8900E+00    6.9060E+00    7.9220E+00    8.9380E+00
   u =     2.2760E-01    2.2755E-01    2.2753E-01    2.2755E-01    2.2760E-01

   x =     8.9380E+00    8.9620E+00    8.9860E+00    9.0100E+00    9.0340E+00
   u =     2.2760E-01    2.4718E-01    2.6699E-01    2.8703E-01    3.0728E-01

   x =     9.0340E+00    1.0158E+01    1.1282E+01    1.2406E+01    1.3530E+01
   u =     3.0728E-01    3.0737E-01    3.0749E-01    3.0766E-01    3.0785E-01

   x =     1.3530E+01    1.3601E+01    1.3671E+01    1.3742E+01    1.3812E+01
   u =     3.0785E-01    4.7950E-01    6.5234E-01    8.2597E-01    1.0000E+00


      x =     0.0000E+00    7.0500E-02    1.4100E-01    2.1150E-01    2.8200E-01
      c =     1.1720E-02    9.6800E-03    7.6451E-03    5.6196E-03    3.6079E-03
  wtper =     6.6400E-01    5.4844E-01    4.3315E-01    3.1839E-01    2.0442E-01

      x =     2.8200E-01    1.4060E+00    2.5300E+00    3.6540E+00    4.7780E+00
      c =     4.0232E-03    4.0206E-03    4.0185E-03    4.0168E-03    4.0156E-03
  wtper =     4.4211E+00    4.4182E+00    4.4159E+00    4.4141E+00    4.4128E+00

      x =     4.7780E+00    4.8020E+00    4.8260E+00    4.8500E+00    4.8740E+00
      c =     4.0420E-03    3.7757E-03    3.5121E-03    3.2515E-03    2.9939E-03
  wtper =     2.2124E-01    2.0666E-01    1.9224E-01    1.7797E-01    1.6387E-01

      x =     4.8740E+00    5.8900E+00    6.9060E+00    7.9220E+00    8.9380E+00
      c =     2.9744E-03    2.9737E-03    2.9735E-03    2.9737E-03    2.9744E-03
  wtper =     3.2685E+00    3.2678E+00    3.2676E+00    3.2678E+00    3.2685E+00

      x =     8.9380E+00    8.9620E+00    8.9860E+00    9.0100E+00    9.0340E+00
      c =     2.9939E-03    3.2515E-03    3.5121E-03    3.7757E-03    4.0420E-03
  wtper =     1.6387E-01    1.7797E-01    1.9224E-01    2.0666E-01    2.2124E-01

      x =     9.0340E+00    1.0158E+01    1.1282E+01    1.2406E+01    1.3530E+01
      c =     4.0156E-03    4.0168E-03    4.0185E-03    4.0206E-03    4.0232E-03
  wtper =     4.4128E+00    4.4141E+00    4.4159E+00    4.4182E+00    4.4211E+00

      x =     1.3530E+01    1.3601E+01    1.3671E+01    1.3742E+01    1.3812E+01
      c =     3.6079E-03    5.6196E-03    7.6451E-03    9.6800E-03    1.1720E-02
  wtper =     2.0442E-01    3.1839E-01    4.3315E-01    5.4844E-01    6.6400E-01


water in each layer as grams in a 1 cm*cm cross section of the material
layer =   1   H2O in layer =   2.158E-03    layer =   2   H2O in layer =   1.807E-02
layer =   3   H2O in layer =   3.374E-04    layer =   4   H2O in layer =   1.209E-02
layer =   5   H2O in layer =   3.374E-04    layer =   6   H2O in layer =   1.807E-02
layer =   7   H2O in layer =   2.158E-03    layer =


total amount of water (grams) in a 1 cm*cm cross section of the full material
total amount of H2O =      5.3213E-02
```

```
time step # =    720      time =   7.200E+02 months     dt =   1.000E+00 months
u = scaled solution = c/scale factor,    c = H2O concentration (g/cc)
wtper = (g H2O)/(100 g dry material) = (100 c)/(rho)
scale factor = solwpa * rho / 100
  x =     0.0000E+00   7.0500E-02   1.4100E-01   2.1150E-01   2.8200E-01
  u =     1.0000E+00   8.5290E-01   7.0613E-01   5.6002E-01   4.1492E-01

  x =     2.8200E-01   1.4060E+00   2.5300E+00   3.6540E+00   4.7780E+00
  u =     4.1492E-01   4.1476E-01   4.1462E-01   4.1451E-01   4.1443E-01

  x =     4.7780E+00   4.8020E+00   4.8260E+00   4.8500E+00   4.8740E+00
  u =     4.1443E-01   3.9713E-01   3.8001E-01   3.6308E-01   3.4634E-01

  x =     4.8740E+00   5.8900E+00   6.9060E+00   7.9220E+00   8.9380E+00
  u =     3.4634E-01   3.4630E-01   3.4629E-01   3.4630E-01   3.4634E-01

  x =     8.9380E+00   8.9620E+00   8.9860E+00   9.0100E+00   9.0340E+00
  u =     3.4634E-01   3.6308E-01   3.8001E-01   3.9713E-01   4.1443E-01

  x =     9.0340E+00   1.0158E+01   1.1282E+01   1.2406E+01   1.3530E+01
  u =     4.1443E-01   4.1451E-01   4.1462E-01   4.1476E-01   4.1492E-01

  x =     1.3530E+01   1.3601E+01   1.3671E+01   1.3742E+01   1.3812E+01
  u =     4.1492E-01   5.6002E-01   7.0613E-01   8.5290E-01   1.0000E+00


    x =     0.0000E+00   7.0500E-02   1.4100E-01   2.1150E-01   2.8200E-01
    c =     1.1720E-02   9.9956E-03   8.2755E-03   6.5633E-03   4.8627E-03
wtper =     6.6400E-01   5.6632E-01   4.6887E-01   3.7186E-01   2.7551E-01

    x =     2.8200E-01   1.4060E+00   2.5300E+00   3.6540E+00   4.7780E+00
    c =     5.4224E-03   5.4202E-03   5.4184E-03   5.4170E-03   5.4160E-03
wtper =     5.9587E+00   5.9563E+00   5.9543E+00   5.9528E+00   5.9517E+00

    x =     4.7780E+00   4.8020E+00   4.8260E+00   4.8500E+00   4.8740E+00
    c =     5.4516E-03   5.2240E-03   4.9988E-03   4.7761E-03   4.5559E-03
wtper =     2.9839E-01   2.8593E-01   2.7361E-01   2.6142E-01   2.4937E-01

    x =     4.8740E+00   5.8900E+00   6.9060E+00   7.9220E+00   8.9380E+00
    c =     4.5262E-03   4.5257E-03   4.5255E-03   4.5257E-03   4.5262E-03
wtper =     4.9738E+00   4.9733E+00   4.9731E+00   4.9733E+00   4.9738E+00

    x =     8.9380E+00   8.9620E+00   8.9860E+00   9.0100E+00   9.0340E+00
    c =     4.5559E-03   4.7761E-03   4.9988E-03   5.2240E-03   5.4516E-03
wtper =     2.4937E-01   2.6142E-01   2.7361E-01   2.8593E-01   2.9839E-01

    x =     9.0340E+00   1.0158E+01   1.1282E+01   1.2406E+01   1.3530E+01
    c =     5.4160E-03   5.4170E-03   5.4184E-03   5.4202E-03   5.4224E-03
wtper =     5.9517E+00   5.9528E+00   5.9543E+00   5.9563E+00   5.9587E+00

    x =     1.3530E+01   1.3601E+01   1.3671E+01   1.3742E+01   1.3812E+01
    c =     4.8627E-03   6.5633E-03   8.2755E-03   9.9956E-03   1.1720E-02
wtper =     2.7551E-01   3.7186E-01   4.6887E-01   5.6632E-01   6.6400E-01


water in each layer as grams in a 1 cm*cm cross section of the material
layer =   1   H2O in layer =   2.335E-03    layer =   2   H2O in layer =   2.436E-02
layer =   3   H2O in layer =   4.801E-04    layer =   4   H2O in layer =   1.839E-02
layer =   5   H2O in layer =   4.801E-04    layer =   6   H2O in layer =   2.436E-02
layer =   7   H2O in layer =   2.335E-03    layer =


total amount of water (grams) in a 1 cm*cm cross section of the full material
total amount of H2O =     7.2748E-02
```

time step # = 960 time = 9.600E+02 months dt = 1.000E+00 months
u = scaled solution = c/scale factor, c = H2O concentration (g/cc)
wtper = (g H2O)/(100 g dry material) = (100 c)/(rho)
scale factor = solwpa * rho / 100

x =	0.0000E+00	7.0500E-02	1.4100E-01	2.1150E-01	2.8200E-01
u =	1.0000E+00	8.7561E-01	7.5150E-01	6.2795E-01	5.0525E-01
x =	2.8200E-01	1.4060E+00	2.5300E+00	3.6540E+00	4.7780E+00
u =	5.0525E-01	5.0511E-01	5.0499E-01	5.0490E-01	5.0483E-01
x =	4.7780E+00	4.8020E+00	4.8260E+00	4.8500E+00	4.8740E+00
u =	5.0483E-01	4.9019E-01	4.7570E-01	4.6136E-01	4.4719E-01
x =	4.8740E+00	5.8900E+00	6.9060E+00	7.9220E+00	8.9380E+00
u =	4.4719E-01	4.4716E-01	4.4715E-01	4.4716E-01	4.4719E-01
x =	8.9380E+00	8.9620E+00	8.9860E+00	9.0100E+00	9.0340E+00
u =	4.4719E-01	4.6136E-01	4.7570E-01	4.9019E-01	5.0483E-01
x =	9.0340E+00	1.0158E+01	1.1282E+01	1.2406E+01	1.3530E+01
u =	5.0483E-01	5.0490E-01	5.0499E-01	5.0511E-01	5.0525E-01
x =	1.3530E+01	1.3601E+01	1.3671E+01	1.3742E+01	1.3812E+01
u =	5.0525E-01	6.2795E-01	7.5150E-01	8.7561E-01	1.0000E+00

x =	0.0000E+00	7.0500E-02	1.4100E-01	2.1150E-01	2.8200E-01
c =	1.1720E-02	1.0262E-02	8.8072E-03	7.3593E-03	5.9213E-03
wtper =	6.6400E-01	5.8140E-01	4.9899E-01	4.1696E-01	3.3549E-01
x =	2.8200E-01	1.4060E+00	2.5300E+00	3.6540E+00	4.7780E+00
c =	6.6029E-03	6.6010E-03	6.5995E-03	6.5983E-03	6.5974E-03
wtper =	7.2559E+00	7.2539E+00	7.2522E+00	7.2509E+00	7.2499E+00
x =	4.7780E+00	4.8020E+00	4.8260E+00	4.8500E+00	4.8740E+00
c =	6.6408E-03	6.4481E-03	6.2575E-03	6.0690E-03	5.8826E-03
wtper =	3.6348E-01	3.5294E-01	3.4250E-01	3.3218E-01	3.2198E-01
x =	4.8740E+00	5.8900E+00	6.9060E+00	7.9220E+00	8.9380E+00
c =	5.8441E-03	5.8437E-03	5.8435E-03	5.8437E-03	5.8441E-03
wtper =	6.4221E+00	6.4216E+00	6.4215E+00	6.4216E+00	6.4221E+00
x =	8.9380E+00	8.9620E+00	8.9860E+00	9.0100E+00	9.0340E+00
c =	5.8826E-03	6.0690E-03	6.2575E-03	6.4481E-03	6.6408E-03
wtper =	3.2198E-01	3.3218E-01	3.4250E-01	3.5294E-01	3.6348E-01
x =	9.0340E+00	1.0158E+01	1.1282E+01	1.2406E+01	1.3530E+01
c =	6.5974E-03	6.5983E-03	6.5995E-03	6.6010E-03	6.6029E-03
wtper =	7.2499E+00	7.2509E+00	7.2522E+00	7.2539E+00	7.2559E+00
x =	1.3530E+01	1.3601E+01	1.3671E+01	1.3742E+01	1.3812E+01
c =	5.9213E-03	7.3593E-03	8.8072E-03	1.0262E-02	1.1720E-02
wtper =	3.3549E-01	4.1696E-01	4.9899E-01	5.8140E-01	6.6400E-01

water in each layer as grams in a 1 cm*cm cross section of the material
layer = 1 H2O in layer = 2.485E-03 layer = 2 H2O in layer = 2.967E-02
layer = 3 H2O in layer = 6.009E-04 layer = 4 H2O in layer = 2.375E-02
layer = 5 H2O in layer = 6.009E-04 layer = 6 H2O in layer = 2.967E-02
layer = 7 H2O in layer = 2.485E-03 layer =

total amount of water (grams) in a 1 cm*cm cross section of the full material
total amount of H2O = 8.9265E-02

```
time step # =   1200      time =   1.200E+03 months     dt =   1.000E+00 months
u = scaled solution = c/scale factor,    c = H2O concentration (g/cc)
wtper = (g H2O)/(100 g dry material) = (100 c)/(rho)
scale factor = solwpa * rho / 100
  x =     0.0000E+00    7.0500E-02    1.4100E-01    2.1150E-01    2.8200E-01
  u =     1.0000E+00    8.9481E-01    7.8985E-01    6.8538E-01    5.8162E-01

  x =     2.8200E-01    1.4060E+00    2.5300E+00    3.6540E+00    4.7780E+00
  u =     5.8162E-01    5.8150E-01    5.8140E-01    5.8132E-01    5.8126E-01

  x =     4.7780E+00    4.8020E+00    4.8260E+00    4.8500E+00    4.8740E+00
  u =     5.8126E-01    5.6888E-01    5.5662E-01    5.4450E-01    5.3251E-01

  x =     4.8740E+00    5.8900E+00    6.9060E+00    7.9220E+00    8.9380E+00
  u =     5.3251E-01    5.3248E-01    5.3248E-01    5.3248E-01    5.3251E-01

  x =     8.9380E+00    8.9620E+00    8.9860E+00    9.0100E+00    9.0340E+00
  u =     5.3251E-01    5.4450E-01    5.5662E-01    5.6888E-01    5.8126E-01

  x =     9.0340E+00    1.0158E+01    1.1282E+01    1.2406E+01    1.3530E+01
  u =     5.8126E-01    5.8132E-01    5.8140E-01    5.8150E-01    5.8162E-01

  x =     1.3530E+01    1.3601E+01    1.3671E+01    1.3742E+01    1.3812E+01
  u =     5.8162E-01    6.8538E-01    7.8985E-01    8.9481E-01    1.0000E+00


    x =     0.0000E+00    7.0500E-02    1.4100E-01    2.1150E-01    2.8200E-01
    c =     1.1720E-02    1.0487E-02    9.2568E-03    8.0323E-03    6.8163E-03
wtper =     6.6400E-01    5.9415E-01    5.2446E-01    4.5509E-01    3.8619E-01

    x =     2.8200E-01    1.4060E+00    2.5300E+00    3.6540E+00    4.7780E+00
    c =     7.6008E-03    7.5993E-03    7.5980E-03    7.5970E-03    7.5963E-03
wtper =     8.3526E+00    8.3509E+00    8.3494E+00    8.3483E+00    8.3475E+00

    x =     4.7780E+00    4.8020E+00    4.8260E+00    4.8500E+00    4.8740E+00
    c =     7.6462E-03    7.4832E-03    7.3220E-03    7.1625E-03    7.0049E-03
wtper =     4.1851E-01    4.0959E-01    4.0077E-01    3.9204E-01    3.8341E-01

    x =     4.8740E+00    5.8900E+00    6.9060E+00    7.9220E+00    8.9380E+00
    c =     6.9592E-03    6.9588E-03    6.9587E-03    6.9588E-03    6.9592E-03
wtper =     7.6474E+00    7.6470E+00    7.6469E+00    7.6470E+00    7.6474E+00

    x =     8.9380E+00    8.9620E+00    8.9860E+00    9.0100E+00    9.0340E+00
    c =     7.0049E-03    7.1625E-03    7.3220E-03    7.4832E-03    7.6462E-03
wtper =     3.8341E-01    3.9204E-01    4.0077E-01    4.0959E-01    4.1851E-01

    x =     9.0340E+00    1.0158E+01    1.1282E+01    1.2406E+01    1.3530E+01
    c =     7.5963E-03    7.5970E-03    7.5980E-03    7.5993E-03    7.6008E-03
wtper =     8.3475E+00    8.3483E+00    8.3494E+00    8.3509E+00    8.3526E+00

    x =     1.3530E+01    1.3601E+01    1.3671E+01    1.3742E+01    1.3812E+01
    c =     6.8163E-03    8.0323E-03    9.2568E-03    1.0487E-02    1.1720E-02
wtper =     3.8619E-01    4.5509E-01    5.2446E-01    5.9415E-01    6.6400E-01


water in each layer as grams in a 1 cm*cm cross section of the material
layer =   1   H2O in layer =   2.612E-03    layer =   2   H2O in layer =   3.416E-02
layer =   3   H2O in layer =   7.030E-04    layer =   4   H2O in layer =   2.828E-02
layer =   5   H2O in layer =   7.030E-04    layer =   6   H2O in layer =   3.416E-02
layer =   7   H2O in layer =   2.612E-03    layer =

total amount of water (grams) in a 1 cm*cm cross section of the full material
total amount of H2O =     1.0323E-01
```

Example No. 1: Run with F4/C20/S4/C20/S4/C20/F4 dt = 1 months
F = 3113 glass-epoxy
C = blasa wood
S = G10 glass epoxy (without copper screen)

FORTRAN OUTPUT FILE IN COLUMN FORM: out5col

```
 column output file name is out5col
 input file name is inp5

time step # =      0      time =   0.000E+00 months     dt =   1.000E+00 months
u = scaled solution = c/scale factor,    c = H2O concentration (g/cc)
wtper = (g H2O)/(100 g dry material) = (100 c)/(rho)
scale factor = solwpa * rho / 100

x = location (cm)    u = scaled conc.   H2O conc. (g/cc)    g H2O/100g material

   0.0000E+00          0.0000E+00          0.0000E+00             0.0000E+00
   7.0500E-02          0.0000E+00          0.0000E+00             0.0000E+00
   1.4100E-01          0.0000E+00          0.0000E+00             0.0000E+00
   2.1150E-01          0.0000E+00          0.0000E+00             0.0000E+00
   2.8200E-01          0.0000E+00          0.0000E+00             0.0000E+00
   2.8200E-01          0.0000E+00          0.0000E+00             0.0000E+00
   1.4060E+00          0.0000E+00          0.0000E+00             0.0000E+00
   2.5300E+00          0.0000E+00          0.0000E+00             0.0000E+00
   3.6540E+00          0.0000E+00          0.0000E+00             0.0000E+00
   4.7780E+00          0.0000E+00          0.0000E+00             0.0000E+00
   4.7780E+00          0.0000E+00          0.0000E+00             0.0000E+00
   4.8020E+00          0.0000E+00          0.0000E+00             0.0000E+00
   4.8260E+00          0.0000E+00          0.0000E+00             0.0000E+00
   4.8500E+00          0.0000E+00          0.0000E+00             0.0000E+00
   4.8740E+00          0.0000E+00          0.0000E+00             0.0000E+00
   4.8740E+00          0.0000E+00          0.0000E+00             0.0000E+00
   5.8900E+00          0.0000E+00          0.0000E+00             0.0000E+00
   6.9060E+00          0.0000E+00          0.0000E+00             0.0000E+00
   7.9220E+00          0.0000E+00          0.0000E+00             0.0000E+00
   8.9380E+00          0.0000E+00          0.0000E+00             0.0000E+00
   8.9380E+00          0.0000E+00          0.0000E+00             0.0000E+00
   8.9620E+00          0.0000E+00          0.0000E+00             0.0000E+00
   8.9860E+00          0.0000E+00          0.0000E+00             0.0000E+00
   9.0100E+00          0.0000E+00          0.0000E+00             0.0000E+00
   9.0340E+00          0.0000E+00          0.0000E+00             0.0000E+00
   9.0340E+00          0.0000E+00          0.0000E+00             0.0000E+00
   1.0158E+01          0.0000E+00          0.0000E+00             0.0000E+00
   1.1282E+01          0.0000E+00          0.0000E+00             0.0000E+00
   1.2406E+01          0.0000E+00          0.0000E+00             0.0000E+00
   1.3530E+01          0.0000E+00          0.0000E+00             0.0000E+00
   1.3530E+01          0.0000E+00          0.0000E+00             0.0000E+00
   1.3601E+01          0.0000E+00          0.0000E+00             0.0000E+00
   1.3671E+01          0.0000E+00          0.0000E+00             0.0000E+00
   1.3742E+01          0.0000E+00          0.0000E+00             0.0000E+00
   1.3812E+01          0.0000E+00          0.0000E+00             0.0000E+00


water in each layer as grams in a 1 cm*cm cross section of the material
layer =   1   H2O in layer =   0.000E+00    layer =   2   H2O in layer =   0.000E+00
layer =   3   H2O in layer =   0.000E+00    layer =   4   H2O in layer =   0.000E+00
layer =   5   H2O in layer =   0.000E+00    layer =   6   H2O in layer =   0.000E+00
layer =   7   H2O in layer =   0.000E+00    layer =

total amount of water (grams) in a 1 cm*cm cross section of the full material
total amount of H2O =     0.0000E+00
```

time step # = 240 time = 2.400E+02 months dt = 1.000E+00 months
u = scaled solution = c/scale factor, c = H2O concentration (g/cc)
wtper = (g H2O)/(100 g dry material) = (100 c)/(rho)
scale factor = solwpa * rho / 100

x = location (cm)	u = scaled conc.	H2O conc. (g/cc)	g H2O/100g material
0.0000E+00	1.0000E+00	1.1720E-02	6.6400E-01
7.0500E-02	7.9333E-01	9.2976E-03	5.2677E-01
1.4100E-01	5.8717E-01	6.8814E-03	3.8988E-01
2.1150E-01	3.8200E-01	4.4768E-03	2.5365E-01
2.8200E-01	1.7832E-01	2.0898E-03	1.1840E-01
2.8200E-01	1.7832E-01	2.3303E-03	2.5608E+00
1.4060E+00	1.7808E-01	2.3273E-03	2.5574E+00
2.5300E+00	1.7789E-01	2.3248E-03	2.5547E+00
3.6540E+00	1.7775E-01	2.3229E-03	2.5526E+00
4.7780E+00	1.7765E-01	2.3216E-03	2.5512E+00
4.7780E+00	1.7765E-01	2.3368E-03	1.2791E-01
4.8020E+00	1.5610E-01	2.0534E-03	1.1239E-01
4.8260E+00	1.3482E-01	1.7734E-03	9.7068E-02
4.8500E+00	1.1379E-01	1.4968E-03	8.1926E-02
4.8740E+00	9.3006E-02	1.2234E-03	6.6964E-02
4.8740E+00	9.3006E-02	1.2155E-03	1.3357E+00
5.8900E+00	9.2956E-02	1.2148E-03	1.3349E+00
6.9060E+00	9.2939E-02	1.2146E-03	1.3347E+00
7.9220E+00	9.2956E-02	1.2148E-03	1.3349E+00
8.9380E+00	9.3006E-02	1.2155E-03	1.3357E+00
8.9380E+00	9.3006E-02	1.2234E-03	6.6964E-02
8.9620E+00	1.1379E-01	1.4968E-03	8.1926E-02
8.9860E+00	1.3482E-01	1.7734E-03	9.7068E-02
9.0100E+00	1.5610E-01	2.0534E-03	1.1239E-01
9.0340E+00	1.7765E-01	2.3368E-03	1.2791E-01
9.0340E+00	1.7765E-01	2.3216E-03	2.5512E+00
1.0158E+01	1.7775E-01	2.3229E-03	2.5526E+00
1.1282E+01	1.7789E-01	2.3248E-03	2.5547E+00
1.2406E+01	1.7808E-01	2.3273E-03	2.5574E+00
1.3530E+01	1.7832E-01	2.3303E-03	2.5608E+00
1.3530E+01	1.7832E-01	2.0898E-03	1.1840E-01
1.3601E+01	3.8200E-01	4.4768E-03	2.5365E-01
1.3671E+01	5.8717E-01	6.8814E-03	3.8988E-01
1.3742E+01	7.9333E-01	9.2976E-03	5.2677E-01
1.3812E+01	1.0000E+00	1.1720E-02	6.6400E-01

water in each layer as grams in a 1 cm*cm cross section of the material
layer = 1 H2O in layer = 1.943E-03 layer = 2 H2O in layer = 1.045E-02
layer = 3 H2O in layer = 1.705E-04 layer = 4 H2O in layer = 4.937E-03
layer = 5 H2O in layer = 1.705E-04 layer = 6 H2O in layer = 1.045E-02
layer = 7 H2O in layer = 1.943E-03 layer =

total amount of water (grams) in a 1 cm*cm cross section of the full material
total amount of H2O = 3.0072E-02

time step # = 480 time = 4.800E+02 months dt = 1.000E+00 months
u = scaled solution = c/scale factor, c = H2O concentration (g/cc)
wtper = (g H2O)/(100 g dry material) = (100 c)/(rho)
scale factor = solwpa * rho / 100

x = location (cm)	u = scaled conc.	H2O conc. (g/cc)	g H2O/100g material
0.0000E+00	1.0000E+00	1.1720E-02	6.6400E-01
7.0500E-02	8.2597E-01	9.6800E-03	5.4844E-01
1.4100E-01	6.5234E-01	7.6451E-03	4.3315E-01
2.1150E-01	4.7950E-01	5.6196E-03	3.1839E-01
2.8200E-01	3.0785E-01	3.6079E-03	2.0442E-01
2.8200E-01	3.0785E-01	4.0232E-03	4.4211E+00
1.4060E+00	3.0766E-01	4.0206E-03	4.4182E+00
2.5300E+00	3.0749E-01	4.0185E-03	4.4159E+00
3.6540E+00	3.0737E-01	4.0168E-03	4.4141E+00
4.7780E+00	3.0728E-01	4.0156E-03	4.4128E+00
4.7780E+00	3.0728E-01	4.0420E-03	2.2124E-01
4.8020E+00	2.8703E-01	3.7757E-03	2.0666E-01
4.8260E+00	2.6699E-01	3.5121E-03	1.9224E-01
4.8500E+00	2.4718E-01	3.2515E-03	1.7797E-01
4.8740E+00	2.2760E-01	2.9939E-03	1.6387E-01
4.8740E+00	2.2760E-01	2.9744E-03	3.2685E+00
5.8900E+00	2.2755E-01	2.9737E-03	3.2678E+00
6.9060E+00	2.2753E-01	2.9735E-03	3.2676E+00
7.9220E+00	2.2755E-01	2.9737E-03	3.2678E+00
8.9380E+00	2.2760E-01	2.9744E-03	3.2685E+00
8.9380E+00	2.2760E-01	2.9939E-03	1.6387E-01
8.9620E+00	2.4718E-01	3.2515E-03	1.7797E-01
8.9860E+00	2.6699E-01	3.5121E-03	1.9224E-01
9.0100E+00	2.8703E-01	3.7757E-03	2.0666E-01
9.0340E+00	3.0728E-01	4.0420E-03	2.2124E-01
9.0340E+00	3.0728E-01	4.0156E-03	4.4128E+00
1.0158E+01	3.0737E-01	4.0168E-03	4.4141E+00
1.1282E+01	3.0749E-01	4.0185E-03	4.4159E+00
1.2406E+01	3.0766E-01	4.0206E-03	4.4182E+00
1.3530E+01	3.0785E-01	4.0232E-03	4.4211E+00
1.3530E+01	3.0785E-01	3.6079E-03	2.0442E-01
1.3601E+01	4.7950E-01	5.6196E-03	3.1839E-01
1.3671E+01	6.5234E-01	7.6451E-03	4.3315E-01
1.3742E+01	8.2597E-01	9.6800E-03	5.4844E-01
1.3812E+01	1.0000E+00	1.1720E-02	6.6400E-01

water in each layer as grams in a 1 cm*cm cross section of the material
layer = 1 H2O in layer = 2.158E-03 layer = 2 H2O in layer = 1.807E-02
layer = 3 H2O in layer = 3.374E-04 layer = 4 H2O in layer = 1.209E-02
layer = 5 H2O in layer = 3.374E-04 layer = 6 H2O in layer = 1.807E-02
layer = 7 H2O in layer = 2.158E-03 layer =

total amount of water (grams) in a 1 cm*cm cross section of the full material
total amount of H2O = 5.3213E-02

time step # = 720 time = 7.200E+02 months dt = 1.000E+00 months
u = scaled solution = c/scale factor, c = H2O concentration (g/cc)
wtper = (g H2O)/(100 g dry material) = (100 c)/(rho)
scale factor = solwpa * rho / 100

x = location (cm)	u = scaled conc.	H2O conc. (g/cc)	g H2O/100g material
0.0000E+00	1.0000E+00	1.1720E-02	6.6400E-01
7.0500E-02	8.5290E-01	9.9956E-03	5.6632E-01
1.4100E-01	7.0613E-01	8.2755E-03	4.6887E-01
2.1150E-01	5.6002E-01	6.5633E-03	3.7186E-01
2.8200E-01	4.1492E-01	4.8627E-03	2.7551E-01
2.8200E-01	4.1492E-01	5.4224E-03	5.9587E+00
1.4060E+00	4.1476E-01	5.4202E-03	5.9563E+00
2.5300E+00	4.1462E-01	5.4184E-03	5.9543E+00
3.6540E+00	4.1451E-01	5.4170E-03	5.9528E+00
4.7780E+00	4.1443E-01	5.4160E-03	5.9517E+00
4.7780E+00	4.1443E-01	5.4516E-03	2.9839E-01
4.8020E+00	3.9713E-01	5.2240E-03	2.8593E-01
4.8260E+00	3.8001E-01	4.9988E-03	2.7361E-01
4.8500E+00	3.6308E-01	4.7761E-03	2.6142E-01
4.8740E+00	3.4634E-01	4.5559E-03	2.4937E-01
4.8740E+00	3.4634E-01	4.5262E-03	4.9738E+00
5.8900E+00	3.4630E-01	4.5257E-03	4.9733E+00
6.9060E+00	3.4629E-01	4.5255E-03	4.9731E+00
7.9220E+00	3.4630E-01	4.5257E-03	4.9733E+00
8.9380E+00	3.4634E-01	4.5262E-03	4.9738E+00
8.9380E+00	3.4634E-01	4.5559E-03	2.4937E-01
8.9620E+00	3.6308E-01	4.7761E-03	2.6142E-01
8.9860E+00	3.8001E-01	4.9988E-03	2.7361E-01
9.0100E+00	3.9713E-01	5.2240E-03	2.8593E-01
9.0340E+00	4.1443E-01	5.4516E-03	2.9839E-01
9.0340E+00	4.1443E-01	5.4160E-03	5.9517E+00
1.0158E+01	4.1451E-01	5.4170E-03	5.9528E+00
1.1282E+01	4.1462E-01	5.4184E-03	5.9543E+00
1.2406E+01	4.1476E-01	5.4202E-03	5.9563E+00
1.3530E+01	4.1492E-01	5.4224E-03	5.9587E+00
1.3530E+01	4.1492E-01	4.8627E-03	2.7551E-01
1.3601E+01	5.6002E-01	6.5633E-03	3.7186E-01
1.3671E+01	7.0613E-01	8.2755E-03	4.6887E-01
1.3742E+01	8.5290E-01	9.9956E-03	5.6632E-01
1.3812E+01	1.0000E+00	1.1720E-02	6.6400E-01

water in each layer as grams in a 1 cm*cm cross section of the material
layer = 1 H2O in layer = 2.335E-03 layer = 2 H2O in layer = 2.436E-02
layer = 3 H2O in layer = 4.801E-04 layer = 4 H2O in layer = 1.839E-02
layer = 5 H2O in layer = 4.801E-04 layer = 6 H2O in layer = 2.436E-02
layer = 7 H2O in layer = 2.335E-03 layer =

total amount of water (grams) in a 1 cm*cm cross section of the full material
total amount of H2O = 7.2748E-02

time step # = 960 time = 9.600E+02 months dt = 1.000E+00 months
u = scaled solution = c/scale factor, c = H2O concentration (g/cc)
wtper = (g H2O)/(100 g dry material) = (100 c)/(rho)
scale factor = solwpa * rho / 100

x = location (cm)	u = scaled conc.	H2O conc. (g/cc)	g H2O/100g material
0.0000E+00	1.0000E+00	1.1720E-02	6.6400E-01
7.0500E-02	8.7561E-01	1.0262E-02	5.8140E-01
1.4100E-01	7.5150E-01	8.8072E-03	4.9899E-01
2.1150E-01	6.2795E-01	7.3593E-03	4.1696E-01
2.8200E-01	5.0525E-01	5.9213E-03	3.3549E-01
2.8200E-01	5.0525E-01	6.6029E-03	7.2559E+00
1.4060E+00	5.0511E-01	6.6010E-03	7.2539E+00
2.5300E+00	5.0499E-01	6.5995E-03	7.2522E+00
3.6540E+00	5.0490E-01	6.5983E-03	7.2509E+00
4.7780E+00	5.0483E-01	6.5974E-03	7.2499E+00
4.7780E+00	5.0483E-01	6.6408E-03	3.6348E-01
4.8020E+00	4.9019E-01	6.4481E-03	3.5294E-01
4.8260E+00	4.7570E-01	6.2575E-03	3.4250E-01
4.8500E+00	4.6136E-01	6.0690E-03	3.3218E-01
4.8740E+00	4.4719E-01	5.8826E-03	3.2198E-01
4.8740E+00	4.4719E-01	5.8441E-03	6.4221E+00
5.8900E+00	4.4716E-01	5.8437E-03	6.4216E+00
6.9060E+00	4.4715E-01	5.8435E-03	6.4215E+00
7.9220E+00	4.4716E-01	5.8437E-03	6.4216E+00
8.9380E+00	4.4719E-01	5.8441E-03	6.4221E+00
8.9380E+00	4.4719E-01	5.8826E-03	3.2198E-01
8.9620E+00	4.6136E-01	6.0690E-03	3.3218E-01
8.9860E+00	4.7570E-01	6.2575E-03	3.4250E-01
9.0100E+00	4.9019E-01	6.4481E-03	3.5294E-01
9.0340E+00	5.0483E-01	6.6408E-03	3.6348E-01
9.0340E+00	5.0483E-01	6.5974E-03	7.2499E+00
1.0158E+01	5.0490E-01	6.5983E-03	7.2509E+00
1.1282E+01	5.0499E-01	6.5995E-03	7.2522E+00
1.2406E+01	5.0511E-01	6.6010E-03	7.2539E+00
1.3530E+01	5.0525E-01	6.6029E-03	7.2559E+00
1.3530E+01	5.0525E-01	5.9213E-03	3.3549E-01
1.3601E+01	6.2795E-01	7.3593E-03	4.1696E-01
1.3671E+01	7.5150E-01	8.8072E-03	4.9899E-01
1.3742E+01	8.7561E-01	1.0262E-02	5.8140E-01
1.3812E+01	1.0000E+00	1.1720E-02	6.6400E-01

water in each layer as grams in a 1 cm*cm cross section of the material
layer = 1 H2O in layer = 2.485E-03 layer = 2 H2O in layer = 2.967E-02
layer = 3 H2O in layer = 6.009E-04 layer = 4 H2O in layer = 2.375E-02
layer = 5 H2O in layer = 6.009E-04 layer = 6 H2O in layer = 2.967E-02
layer = 7 H2O in layer = 2.485E-03 layer =

total amount of water (grams) in a 1 cm*cm cross section of the full material
total amount of H2O = 8.9265E-02

time step # = 1200 time = 1.200E+03 months dt = 1.000E+00 months
u = scaled solution = c/scale factor, c = H2O concentration (g/cc)
wtper = (g H2O)/(100 g dry material) = (100 c)/(rho)
scale factor = solwpa * rho / 100

x = location (cm)	u = scaled conc.	H2O conc. (g/cc)	g H2O/100g material
0.0000E+00	1.0000E+00	1.1720E-02	6.6400E-01
7.0500E-02	8.9481E-01	1.0487E-02	5.9415E-01
1.4100E-01	7.8985E-01	9.2568E-03	5.2446E-01
2.1150E-01	6.8538E-01	8.0323E-03	4.5509E-01
2.8200E-01	5.8162E-01	6.8163E-03	3.8619E-01
2.8200E-01	5.8162E-01	7.6008E-03	8.3526E+00
1.4060E+00	5.8150E-01	7.5993E-03	8.3509E+00
2.5300E+00	5.8140E-01	7.5980E-03	8.3494E+00
3.6540E+00	5.8132E-01	7.5970E-03	8.3483E+00
4.7780E+00	5.8126E-01	7.5963E-03	8.3475E+00
4.7780E+00	5.8126E-01	7.6462E-03	4.1851E-01
4.8020E+00	5.6888E-01	7.4832E-03	4.0959E-01
4.8260E+00	5.5662E-01	7.3220E-03	4.0077E-01
4.8500E+00	5.4450E-01	7.1625E-03	3.9204E-01
4.8740E+00	5.3251E-01	7.0049E-03	3.8341E-01
4.8740E+00	5.3251E-01	6.9592E-03	7.6474E+00
5.8900E+00	5.3248E-01	6.9588E-03	7.6470E+00
6.9060E+00	5.3248E-01	6.9587E-03	7.6469E+00
7.9220E+00	5.3248E-01	6.9588E-03	7.6470E+00
8.9380E+00	5.3251E-01	6.9592E-03	7.6474E+00
8.9380E+00	5.3251E-01	7.0049E-03	3.8341E-01
8.9620E+00	5.4450E-01	7.1625E-03	3.9204E-01
8.9860E+00	5.5662E-01	7.3220E-03	4.0077E-01
9.0100E+00	5.6888E-01	7.4832E-03	4.0959E-01
9.0340E+00	5.8126E-01	7.6462E-03	4.1851E-01
9.0340E+00	5.8126E-01	7.5963E-03	8.3475E+00
1.0158E+01	5.8132E-01	7.5970E-03	8.3483E+00
1.1282E+01	5.8140E-01	7.5980E-03	8.3494E+00
1.2406E+01	5.8150E-01	7.5993E-03	8.3509E+00
1.3530E+01	5.8162E-01	7.6008E-03	8.3526E+00
1.3530E+01	5.8162E-01	6.8163E-03	3.8619E-01
1.3601E+01	6.8538E-01	8.0323E-03	4.5509E-01
1.3671E+01	7.8985E-01	9.2568E-03	5.2446E-01
1.3742E+01	8.9481E-01	1.0487E-02	5.9415E-01
1.3812E+01	1.0000E+00	1.1720E-02	6.6400E-01

water in each layer as grams in a 1 cm*cm cross section of the material
layer = 1 H2O in layer = 2.612E-03 layer = 2 H2O in layer = 3.416E-02
layer = 3 H2O in layer = 7.030E-04 layer = 4 H2O in layer = 2.828E-02
layer = 5 H2O in layer = 7.030E-04 layer = 6 H2O in layer = 3.416E-02
layer = 7 H2O in layer = 2.612E-03 layer =

total amount of water (grams) in a 1 cm*cm cross section of the full material
total amount of H2O = 1.0323E-01

DISTRIBUTION

DOD ACTIVITIES (CONUS)

	Copies
ATTN 332 (JAMES J. KELLY)	1
331PO (DR K J WYNNE)	1
332 (DR L E SLOTTER)	1
DIRECTOR	
OFFICE OF NAVAL RESEARCH	
BALLSTON TOWER - 504	
800 N QUINCY STREET	
ARLINGTON VA 22217-5660	
ATTN AIR-52032 (S STEEL)	1
AIR-52032 (D MOOR)	1
COMMANDER	
NAVAL AIR SYSTEMS COMMAND	
WASHINGTON DC 20362	
ATTN SEA-03B	1
SEA-03R (SPENCER)	1
SEA-05R	1
SEA-062D	1
SEA-06K42 (EADIE)	1
SEA-09G32	1
COMMANDER	
NAVAL SEA SYSTEMS COMMAND	
WASHINGTON DC 20362	
ATTN 6380 (DR R BADALIANCE)	1
8433 (DR I WOLOCK)	1
DIRECTOR	
NAVAL RESEARCH LABORATORY	
WASHINGTON DC 20375	
ATTN 824 (LOGAN)	1
COMMANDER	
NAVAL COMMAND, CONTROL, AND OCEAN SURVEILLANCE CENTER	
53560 HULL ST	
SAN DIEGO CA 92152-5001	
ATTN B44 (A E BERGER)	5
F24 (CAMPBELL)	1
COMMANDER	
NAVAL SURFACE WARFARE CENTER	
DAHLGREN DIVISION	
17320 DAHLGREN RD	
DAHLGREN VA 22448-5100	
DIRECTOR OF DEVELOPMENT	
ARMY MATERIAL COMMAND	
GRAVELEY POINT	
WASHINGTON DC 20316	1
ATTN 6043 (E ROSENZWEIG)	1
606 (J DELUCCIA)	1
6063	1
6064 (R TROBACCO)	1
COMMANDER	
NAVAL AIR DEVELOPMENT CENTER	
DEPARTMENT OF THE NAVY	
WARMINSTER PA 18974	
ATTN WL/DOC (TECH LIBRARY)	1
WL/MLBC (C BROWNING)	1
WRIGHT LABORATORY	
WRIGHT PATTERSON AFB OH 45433-7411	
DEFENSE TECHNICAL INFORMATION CENTER	
8725 JOHN J KINGMAN ROAD	
SUITE 09844	
FORT BELVOIR VA 22060-6218	2

NON-DOD ACTIVITIES

NONE

DISTRIBUTION (Continued)

NSWC CARDEROCK DIVISION INTERNAL DISTRIBUTION		Copies
011	(CORRADO)	1
0112	(DOUGLAS)	1
0115	(CAPLAN)	1
0119	(MESSICK)	1
3021	(COLEGROVE)	1
3023	(TIC/CARDEROCK)	1
3023	(TIC/WHITE OAK)	1
3024	(TIC/ANNAPOLIS)	1
60	(WACKER)	1
601	(MORTON)	1
602	(ROCKWELL)	1
603	(CAVALLARO)	1
61	(HOLSBERG)	1
64	(FISCHER)	1
6401	(CASTELLI)	1
641	(HOWELL)	1
643	(SORATHIA)	1
6501	(BENSON)	1
655	(CRITCHFIELD)	1
6551	(MAYES)	1
6552	(GIPPLE)	1
6552	(LOUP)	1
6552	(TELEGADAS)	1
6553	(COFFIN)	1
6553	(JUSKA)	1
6553	(WILLIAMS)	2
68	(MUELLER)	1
681	(AUGL)	15
681	(HARTMANN)	1
681	(J D LEE)	1
823	(WILHELMI)	1